

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:

INGENIERÍA DE SISTEMAS

**Trabajo de titulación previo a la obtención del título: INGENIERO DE
SISTEMAS**

TEMA:

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB
PARA LA ADMINISTRACIÓN Y GESTIÓN DE PARTES OPERATORIOS EN
QUIRÓFANOS DEL HOSPITAL DE ESPECIALIDADES EUGENIO ESPEJO**

AUTORES:

**DIEGO ARMANDO CANCHIGNIA TERÁN
MANUEL EDUARDO MOINA CAMPOS**

DIRECTOR:

ALONSO RENÉ ARÉVALO CAMPOS

Quito, febrero de 2015

DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE TITULACIÓN

Nosotros autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Quito, febrero de 2015

Diego Armando Canchignia Terán

1721998019

Manuel Eduardo Moina Campos

1719139246

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	1
DIAGNÓSTICO DE LA SITUACIÓN	2
1.1. Objetivos	3
1.1.1. Objetivo General	3
1.1.2. Objetivos Específicos.....	3
1.2. Justificación.....	3
1.3. Marco Teórico	4
1.3.1. Metodología de desarrollo de software	4
1.3.1.1. Scrum	4
1.3.1.1.1. Componentes de Scrum.....	5
1.3.2. Arquitectura de la aplicación web.....	8
1.3.2.1. Plataforma JEE.....	8
1.3.2.1.1. Java Server Faces (JSF)	10
1.3.2.1.2. Enterprise Java Bean (EJB).....	13
1.3.2.1.3. Java Persistence API (JPA).....	16
1.3.2.2. Primefaces	17
1.3.2.2.1. Justificativo de utilizar este Framework	18
1.3.2.3. GlassFish	20
1.3.2.3.1 Justificativo de utilizar GlassFish	20
1.3.2.4. MySQL.....	21
1.3.2.4.1. Características de MySQL	22
CAPÍTULO 2	23
ANÁLISIS Y DISEÑO DEL SISTEMA INFORMÁTICO	23
2.1. Especificación de requerimiento	27
2.1.1. Recopilación de información para la pila de producto.....	30
2.1.1.1. Historia de usuario	30
2.1.1.2. Product Backlog (Pila de producto)	31
2.2. Planificación del desarrollo.....	33
2.2.1. Sprint Planning (Planificación de las Iteraciones)	33

2.2.1.1. Sprint 1	34
2.2.1.1.1. Tarea 1 (Control de acceso)	34
2.2.1.1.2. Inconveniente en la tarea 1 (Control de acceso)	36
2.2.1.1.3. Tarea 1 terminada (Control de acceso)	38
2.2.1.1.4. Daily Scrum (Control de acceso)	39
2.2.1.1.5. Sprint Review.....	40
2.2.1.1.6. Sprint Retrospective	40
2.3. Diseño de diagramas	41
2.3.1. Diagrama de casos de uso	41
2.3.1.1. Control de acceso	42
2.3.1.2. Definición de roles	44
2.3.2. Diagramas de secuencia	45
2.3.2.1. Control de acceso	46
2.3.2.2. Definiciones de roles.....	46
2.3.3. Diagramas de clase.....	47
2.3.3.1. Diagrama de clases (Controladores)	48
2.4. Diseño de la Base de Datos	49
2.4.1. Diseño Lógico	49
2.4.2. Diseño Físico.....	51
2.4.3.1. Usuarios	52
2.4.3.2. Perfil	52
2.4.3.3. Opción por perfil	53
2.4.3.4. Opción	53
2.5. Diseño de la interfaz de la aplicación	54
2.5.1. Pantalla de logeo para el control de acceso.....	54
2.5.2. Menú por perfiles	54
2.5.3. Submenú desplegable en menú de perfiles	55
2.5.4. Registros usuarios, perfiles, profesionales	55
2.5.5. Submenú desplegable en registros de usuarios, perfiles, profesionales.....	56
2.5.6. Modificaciones de registros de usuarios, perfiles, profesionales	56
2.5.7. Submenú desplegable en modificaciones de registros	57

2.5.8. Detalle de registros.....	57
CAPÍTULO 3	58
GENERACIÓN DEL SISTEMA INFORMÁTICO	58
3.1. Método Esquemático.....	58
3.2. Diagrama de proceso del Parte Operatorio	59
3.3. Desarrollo del código de la aplicación	60
3.3.1. Control de acceso	60
3.3.2. Definición de roles	64
3.3.2.1. Página principal de administración	64
3.3.2.2. Administración de usuarios.....	65
3.3.2.3. Registro de usuarios y definición de roles	65
3.3.3. Registro de profesionales	66
3.3.3.1. Página principal de registro de profesionales.....	66
3.3.3.2. Registro de profesionales.	67
3.3.4. Pedido de cirugía.....	68
3.3.4.1. Página principal de pedido de cirugía	68
3.3.4.2. Registro del pedido de cirugía.....	68
3.3.5. Aprobación de solicitud de cirugía	69
3.3.5.1. Pantalla principal de solicitud de cirugía	69
3.3.5.2. Aprobación de solicitud de cirugía	70
CAPÍTULO 4	72
IMPLANTACIÓN	72
4.1. Pruebas de seguridad de acceso	72
4.1.1. Inyección SQL	72
4.1.2. Consola web	73
4.2. Pruebas de funcionalidad y rendimiento	75
4.2.1. Funcionalidad.....	75
4.2.1.1. Plan de pruebas 1	75
4.2.1.2. Plan de pruebas 2	77
4.2.1.3. Plan de pruebas 3	79
4.3. Implantación.....	85

CONCLUSIONES

RECOMENDACIONES

LISTA DE REFERENCIAS

ÍNDICE DE TABLAS

Tabla 1. Tabla de especificaciones JEE	9
Tabla 2. Comparación de Frameworks	19
Tabla 3. Comparación entre GlassFish y Tomcat	21
Tabla 5. Historia de usuario 2	31
Tabla 6. Pila de producto, registra los requerimientos para la aplicación web	32
Tabla 7. Sprint 1	34
Tabla 8. Tarea 1 del control de acceso	35
Tabla 9. Tarea 1 terminada del control de acceso	38
Tabla 10. Daily Scrum del control de acceso	39
Tabla 11. Sprint Review del Sprint 1	40
Tabla 12. Sprint Retrospective del Sprint 1	41
Tabla 13. Tabla del diagrama de casos de uso de la definiciones de roles	44
Tabla 14. Tabla del diagrama de casos de uso del control de acceso	42
Tabla 15. Tabla del diccionario de datos de usuario	52
Tabla 16. Tabla del diccionario de datos de perfil	52
Tabla 17. Tabla del diccionario de datos de opción de perfil	53
Tabla 18. Tabla del diccionario de datos de opción	53
Tabla 19. Plan de prueba 1	77
Tabla 20. Plan de prueba 2	79
Tabla 21. Plan de pruebas 3	81

ÍNDICE DE FIGURAS

Figura 1. Proceso de Scrum	4
Figura 2. Arquitectura de Java Enterprise Edition 6	10
Figura 3. Ciclo de vida de JSF	12
Figura 4. Registro de EJB	13
Figura 5. Funcionamiento de los componentes EJB	15
Figura 6. Arquitectura de JPA.....	17
Figura 7. Diagrama de proceso,	29
Figura 8. Pizarra de Sprint	33
Figura 9. Tarea 1 del control de acceso.....	36
Figura 10. Base de datos sin tabla para roles	37
Figura 11. Diagrama de casos de uso del control de acceso	43
Figura 12. Diagrama de casos de uso de la definiciones de roles	45
Figura 13. Diagrama de secuencia de control de acceso.....	46
Figura 14. Diagrama de secuencia de definición de roles.....	46
Figura 15. Diagrama de clases	47
Figura 16. Diagrama de clases de controladores 1	48
Figura 17. Diagrama de clases de controladores 2.....	49
Figura 18. Diagrama lógico de la base de datos.....	50
Figura 19. Diagrama físico de la base de datos.....	51
Figura 20. Pantalla de logeo.....	54
Figura 21. Pantalla de menús por perfiles	54
Figura 22. Pantalla con submenú desplegable en menú perfiles.....	55
Figura 23. Pantalla de registros	55
Figura 24. Pantalla submenú desplegable en registro	56
Figura 25. Pantalla de modificaciones	56
Figura 26. Pantalla submenú desplegable en modificaciones de registros	57
Figura 27. Pantalla de detalle de registros	57
Figura 28. Diagrama esquemático	58
Figura 29. Diagrama de procesos del parte operatorio	60

Figura 30. Interfaz de control de acceso	61
Figura 31. Interfaz principal de administración	64
Figura 32. Interfaz de administración de usuarios	65
Figura 33. Interfaz de registro de usuario	65
Figura 34. Interfaz principal de registro de profesionales.....	66
Figura 35. Interfaz de registro de profesionales.....	67
Figura 36. Interfaz principal de pedido de cirugía	68
Figura 37. Interfaz de registro de pedido de cirugía	68
Figura 38. Interfaz principal de pedido de cirugía	69
Figura 39. Interfaz de aprobación de pedido de cirugía.....	70
Figura 40. Consulta en la base de datos con usuario.....	72
Figura 41. Inyección SQL en base de datos con usuario	73
Figura 42. Inyección SQL en la aplicación web	73
Figura 43. Prueba con inspector.....	74
Figura 44. Prueba con consola	74
Figura 45. Prueba con consola usuario correcto	75
Figura 46. Recuperación de contraseña	76
Figura 47. Administrador de la aplicación.....	78
Figura 48. Nuevo usuario.....	78
Figura 49. Ingreso con usuario nuevo	79
Figura 50. Ingreso usuario secretaría de quirófano.....	80
Figura 51. Registro de Hoja de Anestesia.....	80
Figura 52. Verificación del registro del parte operatorio.....	81
Figura 53. Jmeter petición HTTP.....	82
Figura 54. Jmeter Ver Resultado en Árbol	82
Figura 55. Jmeter Response Time Graph.....	83
Figura 56. Jmeter Ver Árbol de Resultados	83
Figura 57. Jmeter Gráficos de Resultados.....	84
Figura 58. Jmeter Visualizador Spline	84
Figura 59. Jmeter Gráfico	85

ÍNDICE DE ANEXOS

Anexo 1. Manual de Instalación	87
Anexo 2. Manual de Usuario	93
Anexo 3. Documento de Parte Operatorio del HEE	100
Anexo 3. Cuadros estadísticos de cirugías	102
Anexo 4. Actas de entrega de Sprints	103
Anexo 5. Diamagra de procesos del Parte Operatorio aprobado	107
Anexo 6. Documento de auspicio de tesis del HEE.....	108
Anexo 4. Documento de entrega y finalización de tesis HEE	109

RESUMEN

En cooperación mutua entre la entidad pública Hospital de Especialidades Eugenio Espejo, bajo la Subdirección de Docencia e Investigación y los tesisistas de la Universidad Politécnica Salesiana, se firmó un auspicio, con el cual los estudiantes desarrollaron una aplicación web para administrar Partes Operatorios. La gestión del proyecto se llevó a cabo utilizando la metodología de desarrollo de software Scrum.

Scrum es una metodología ágil que permite la administración de iteraciones o entregas programadas, que se adaptan a los cambios en el desarrollo.

El desarrollo de la aplicación web cuenta con tres Sprints: El Sprint 1 y 2 están previstos para 30 días y el Sprint 3 para 20.

El **Sprint 1** cuenta con:

- Control de Acceso
- Definición de Roles
- Registro de Profesionales

El **Sprint 2** cuenta con:

- Pedido de Cirugía
- Aprobación de Solicitud
- Registro Parte Operatorio

El **Sprint 3** cuenta con:

- Publicaciones
- Informes

La arquitectura de la aplicación web está basada en la plataforma JEE, JSF integra el Framework PrimeFaces, se usó EJB que es un contenedor de objetos java, mediante JPA que es la persistencia para la comunicación con la base de datos, ayudaron a desarrollar rápidamente el proyecto, el gestor de base de datos utilizado es MySQL.

Las pruebas de funcionalidad están establecidas de acuerdo a los casos de uso, las pruebas de control de acceso están validadas para no admitir inyección SQL mediante JavaScript y además las contraseñas cuentan con encriptación MD5 y SHA1, las pruebas de rendimientos están efectuadas mediante la herramienta Jmeter (propiedad de Apache).

ABSTRACT

In mutual cooperation between the public entity Hospital of specialties Eugenio Espejo, under the Office of teaching and research and the thesis students of the Salesian Polytechnic University, signed a sponsorship, with which students developed a web application to manage operative parts. The management of the project was carried out using the methodology of development of software Scrum.

Scrum is an agile methodology that allows the administration of iterations or scheduled deliveries, that adapt to changes in the development.

The development of the web application has three Sprints: The Sprint 1 and 2 are planned for 30 days and the Sprint 3 to 20.

The Sprint 1 has:

- Access Control
- Definition of Roles
- Professionals' record

The Sprint 2 has:

- Order of Surgery
- Approval Request
- Register Part operative

The Sprint 3 has:

- Publications
- Reports

The architecture of the Web application is based on the platform JEE, JSF integrates the Framework PrimeFaces, is use EJB which is a container objects java, by JPA which is the persistence for communication with the database, helped quickly develop the project, the manager database used is MySQL.

tests functionality were established according to use cases, tests access control were validated for not admit SQL injection by JavaScript, and also passwords have encryption MD5 and SHA1, tests yields are made by the tool Jmeter (owned by Apache).

INTRODUCCIÓN

Actualmente el Hospital de Especialidades Eugenio Espejo cuenta con un parque informático capaz de resolver los requerimientos que se tenga en el área de tecnología.

Uno de los problemas a solucionar según los requerimientos levantados, es el registro en los partes operatorios, para solventar este requerimiento se propone una solución web, la misma que operará en la intranet de la institución.

La aplicación web permitirá administrar la información del parte operatorio y facilitará la obtención de reportes parametrizados según los requerimientos de la institución. Es importante recalcar que la institución no cuenta con un sistema informático que permita realizar el proceso.

La información que administrará el sistema tendrá 2 fuentes de datos:

- a) La base de datos de pacientes y especialistas con la que cuenta la institución
- b) La base de datos a diseñarse para la aplicación a desarrollarse

CAPÍTULO 1

DIAGNÓSTICO DE LA SITUACIÓN

El Hospital de Especialidades Eugenio Espejo es una casa de salud muy representativa de Ecuador, y actualmente está dentro de los establecimientos más grandes del país. Cuenta con un aproximado de 2500 colaboradores estimado a la fecha, de los cuales 300 son médicos especializados (Nomina Abril de empleados HEE 2014, 2014). Atienden alrededor de 1500 pacientes diariamente, pueden ser ecuatorianos o extranjeros (Estadística de pacientes mensuales HEE 2014, 2014).

- Durante años en el Hospital de Especialidades Eugenio Espejo, la gestión de partes operatorios se ha realizado manualmente. Como resultado de esto, existe duplicación y pérdida de información, esto implica procesos lentos y tediosos, tanto para el personal de área y solicitantes (médicos especialistas, enfermeras de piso, Auditoría Interna y Estadística). Esto provoca, la falta de control de los medicamentos e implementos ocupados en quirófanos.
- Los problemas detectados en relación con los partes operatorios en el Hospital de Especialidades Eugenio Espejo son los siguiente:
 - Impresión del parte operatorio físico
 - Registro manual del parte operatorio
 - Falta de políticas de restricción a la información en el parte operatorio
- Consecuencias:
 - Impresión desmedida de hojas
 - Perdida del parte operatorio
 - Deterioro del parte operatorio por el paso del tiempo
 - Registros lentos
 - Mal agendamiento de especialistas
 - Duplicación de información
 - Información poco legible
 - Falla en búsquedas y estadísticas
 - Cambio en información registrada en el parte operatorio archivado

1.1. Objetivos

1.1.1. Objetivo General

Desarrollar un sistema informático capaz de administrar los partes operatorias en quirófanos del Hospital de Especialidades Eugenio Espejo, esto relacionado con los días y horarios para operaciones de cirujanos además el personal de apoyo (enfermeras, anestesiólogos, auxiliares de enfermería).

1.1.2. Objetivos Específicos

1. Agendar pacientes que serán intervenidos quirúrgicamente
2. Agendar responsables o personal que intervendrán en la cirugía
3. Registrar un catálogo de especialistas, y personal de apoyo para las cirugías
4. Registrar un catálogo de especialidades
5. Registrar observaciones de la cirugía
6. Generar el parte operatorio
7. Generar reportes estadísticos de cirugías, por especialidad y cirugías exitosas
8. Implantar sistema piloto

1.2. Justificación

Las demandas actuales en los quirófanos requieren de formas nuevas, amigables y flexibles de software, que permitan generar soluciones eficaces y/o adecuadas al personal que labora en este sitio de trabajo.

Implementar el sistema informático piloto funcional en los quirófanos del Hospital de Especialidades Eugenio Espejo para automatizar el manejo de partes operatorios y mejorar el manejo de información de cirugías.

Con el objetivo de mejorar los trámites de gestión de partes operatorios se plantea una solución web, teniendo en cuenta que en el Hospital de Especialidades Eugenio Espejo,

existen diferentes plataformas de sistemas operativos en los equipos informáticos de los usuarios, sin tener una aplicación que realice dichos parte operatorios se realizará un agendamiento cruzado, manejo de perfiles y registro de información del parte operatorio.

1.3. Marco Teórico

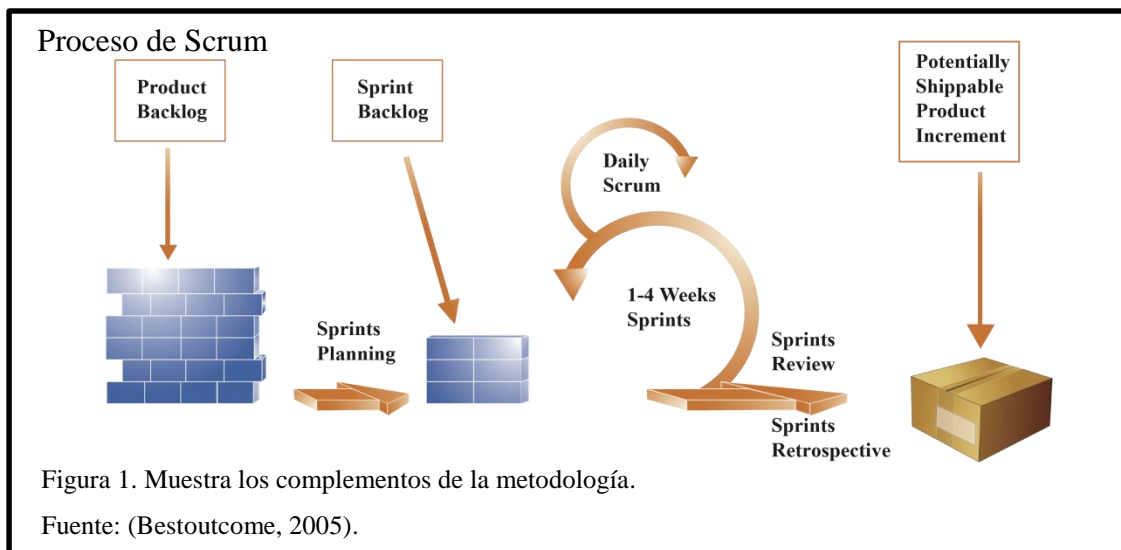
Este marco teórico ayudará a fundamentar los conceptos en los que se basa las herramientas que se ocuparán en este proyecto.

1.3.1. Metodología de desarrollo de software

1.3.1.1. Scrum

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa (ROI). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación.

Este método fue desarrollado por Ken Schwaber, Jeff Sutherland, Mike Beedle. Esta es una herramienta conocida como hiperproductividad, la misma que fue pulida y reforzada.



1.3.1.1.1. Componentes de Scrum

Artefactos

- **Product Backlog (Pila de producto):** Son los requerimientos del producto. Elaborado por: el Product Owner y priorizada según la importancia para el proyecto. Es lo que se va hacer.
- **Sprint Backlog (Ítems del Product Backlog):** Listado escogido por el equipo de trabajo, para ejecutarse durante el Sprint seleccionado. El equipo establece la duración de cada Sprint.

Reuniones

- **Sprint Planning (Planificación de Sprint):** Define y/o planifica cómo se enfocará el proyecto, prioriza los requerimientos del producto, están todas las etapas y los plazos, según la necesidad y la brevedad de cada sprint.
- **Daily Scrum Meeting (Reunión diaria):** Reunión que se realiza a diario de manera breve (Stand-up meeting) durante el Sprint. Se revisa lo qué se hizo ayer, lo que se va hacer hoy, y si hay algún problema como lo resuelvo, para eso está el apoyo del Scrum Master que procurará solucionar problemas u obstáculos que se presenten durante el camino.
- **Sprint Review (Revisión del sprint terminado):** Revisión del piloto listo para ser presentado al cliente.
- **Sprint Retrospective (Revisión en Retrospectiva del Sprint):** Revisión de los objetivos que se han cumplidos en la iteración terminada. Se registra los problemas y soluciones para no volver a repetir los errores. Ayuda a mejorar el proceso del desarrollo.

Participantes

- **Product Owner (Intermediario):** Capta las ideas del cliente y transcribe los requerimientos, asegura que el equipo de trabajo cumpla los objetivos y las expectativas. Es igual que un jefe o responsable del proyecto, puede ser una persona interna o externa de la institución.

- **Scrum Master (Líder de reuniones):** Ayuda al equipo de trabajo cuando tienen problemas, minimiza obstáculos, pero hay que tener claro que no es un gestor, sino es un apoyo que facilita las cosas.
- **Scrum Team (Grupo de trabajo):** Desarrolladores que cumplen lo asignada del Product Owner.
- **Cliente:** Dueño del producto, tiene influencia directa en el proceso y el producto resultante.

Beneficios

- **Cumplimento de expectativas:** El cliente establece sus expectativas indicando el valor que le aporta cada requisito / **historia** del proyecto, el equipo los estima y con esta información el **Product Owner** establece su prioridad. De manera regular, en las demos de Sprint el **Product Owner** comprueba que efectivamente los requisitos se han cumplido y transmite se feedback al equipo.
- **Flexibilidad a cambios:** Alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos.
- **Reducción del Time to Market:** El cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- **Mayor calidad del software:** La necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.
- **Mayor productividad:** Se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.
- **Maximiza el retorno de la inversión (ROI):** Muestra el costo de una actividad, con mayores tareas, el costo de la actividad baja.
- **Predicciones de tiempos:** Mediante esta metodología se conoce la velocidad media del equipo por sprint (los llamados puntos historia), con lo que

consecuentemente, es posible estimar fácilmente para cuando se dispondrá de una determinada funcionalidad que todavía está en el Backlog.

- **Reducción de riesgos:** El hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada (web softeng.es, 2013).

¿Cuándo o porque se utiliza?

Cuando necesitamos presentar un producto en el menor tiempo posible, teniendo cambios en el transcurso del desarrollo, y obligados a presentar módulos funcionales al cliente.

Se basa en iteraciones e incrementos los cuales deben estar bien definidos, se ha optado por esta metodología para el desarrollo del sistema.

Es aplicable al requerimiento ya que se necesita entregar adelantos del sistema funcional y luego se realizará la documentación, y teniendo en cuenta los sprints o tiempos para entrega de producto funcional por iteraciones y así incrementando su desarrollo se adapta a una fecha de entrega.

Por sus características de adaptarse a las necesidades de los desarrolladores y a los usuarios, se necesita solo una reunión global con el cliente y una diaria con el Scrum Team para fijar fechas y revisar los avances para que no se vuelva algo tedioso de parte y parte.

En este proyecto como product owner no se necesita recopilar información o requerimientos, se cuenta con información y requerimientos claros y concisos que necesitan ser revisados e impresos en las iteraciones, dando así procesos claros, e ideas en cada progreso, también solo los involucrados darán su opinión.

Facilita pruebas en el software de manera rápida y adaptativa a cambios. Con Scrum se evita la generación de documentación que no es necesaria por cambios en el desarrollo del sistema y evitar la burocracia.

1.3.2. Arquitectura de la aplicación web

1.3.2.1. Plataforma JEE

Java Platform, Enterprise Edition o Java EE, traducido informalmente como Java Empresarial, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

Java EE tiene varias especificaciones de API², tales como JDBC³, Servicios Web, etc y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets, JavaServerPages (JSF) y varias tecnologías de servicios web. Ello permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel (web java.com, 2014, pág. 8).

Tabla 1.

Tabla de especificaciones JEE

PLATAFORMA JAVA EE	
Tecnologías de Aplicación Web	
JSR 315:	Java Servlet 3.0
JSR 314:	Java Server Faces (JSF) 2.6
JSR 245:	Java Server Pages 2.2 and Expression Language (1.2)
JSR-52:	A Standard Tag Library for Java Server Pages 1.2
JSR-45:	Debugging Support for Other Languages 1.0
Tecnologías de Aplicación Enterprise	
JSR 299:	Context and Dependency Injection for the Java EE Platform 1.0
JSR 330:	Dependency Injection for Java
JSR 318:	Enterprise Java Beans 3.1
JSR 317:	Java Persistence API 2.0
JSR 250:	Common Annotations for the Java Platform 1.1
JSR 907:	Java Transaction API (JTA) 1.1
JSR 303:	Bean Validation 1.0
JSR 322:	Java EE Connector Architecture 1.6
JSR 914:	Java Message Service (JMS) API 1.1
JSR 919:	Java Mail 1.4
Tecnologías de Servicios Web	
JSR 311:	JAX-RS: The Java API for REST ful Web Services 1.1
JSR 109:	Implementing Enterprise Web Services 1.3
JSR 224:	Java API for XML-Based Web Services (JAX-WS) 2.2
JSR 181:	Web Services Metadata for the Java Platform
JSR 101:	Java APIS for XML Based RPC 1.1
JSR 93:	Java APIS for XML Registries 1.0 (JAXR) 1.0
Tecnologías de Administración y Seguridades	
JSR 196:	Java Authentication Service Provider Interface for Containers 1.0
JSR 115:	Java Authorization Contract for Containers 1.0
JSR 77:	J2EE Management 1.1
JSR 88:	Java EE Application Deployment 1.2

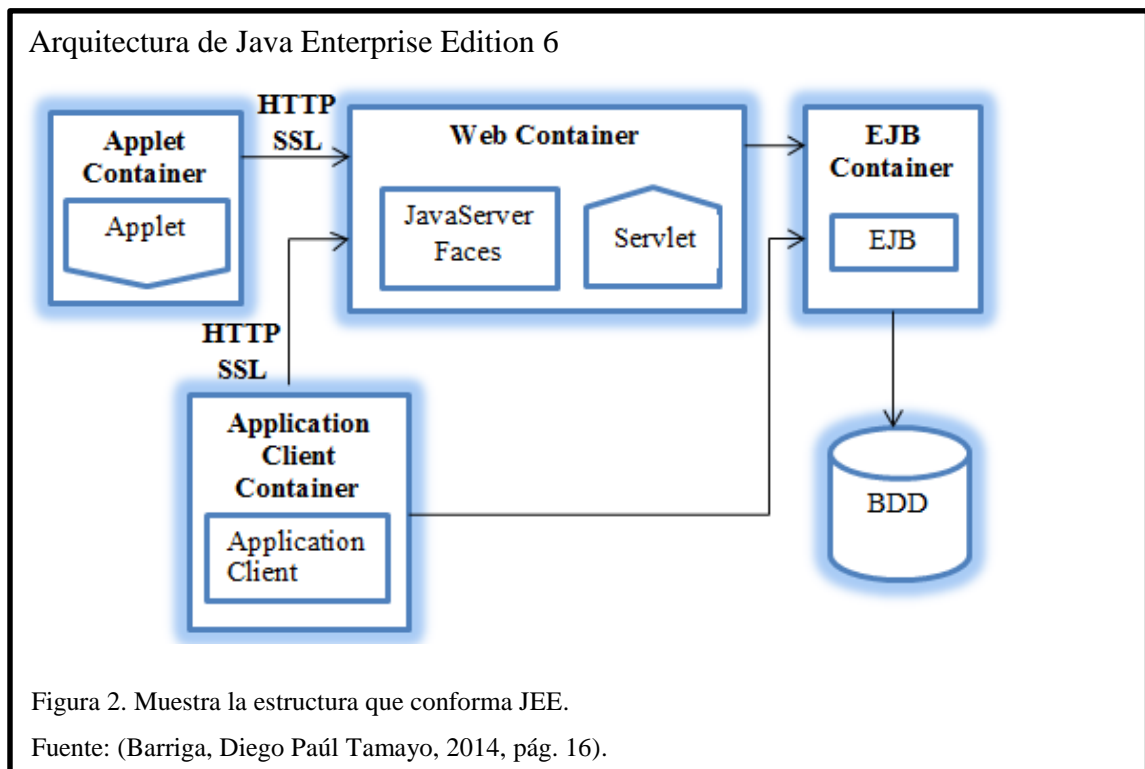
Nota: Especifica la versión 6 y las especificaciones de la tecnología de la plataforma

Fuente: (Docs Jboss, 2014).

JEE utiliza el apoyo de componentes de software modulares que se ejecutan en un mismo server de aplicaciones porque cuenta con arquitectura distribuida de N capas. También configura especificaciones únicas para componentes, entre ellas estan EJB, Servlet, y tecnologías de servicio web. Ello permite crear una aplicación empresarial portable para las diferentes plataformas del mercado informático. Esto permite que se pueda ejecutar y desplegar en cualquier servidor web o de aplicaciones que cumpla con el estándar requerido.

2. **API** (*Application Programming Interface*) es una serie de servicios o funciones que ofrece una librería JAVA al programador.

3. **JDBC** (Java Database Connectivity) es una interfaz que permite a un programa java ejecutar instrucciones SQL dentro de bases de datos.



Los servers tienen un middleware el cual permite intercomunicarse con varios servicios para la seguridad y confiabilidad de la información. Los servers de aplicación brindan a los desarrolladores una interfaz para programación de aplicaciones, de tal manera que no tenga que preocuparse por el sistema operativo o por la gran cantidad de interfaces requeridas en una aplicación web moderna (Barriga, Diego Paúl Tamayo, 2014, pág. 20).

1.3.2.1.1. Java Server Faces (JSF)

JavaServer Faces es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServerPages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL (acrónimo de XML-based User-interface Language, lenguaje basado en XML para la interfaz de usuario).

JavaServer Faces maneja el estado de los componentes de interfaz de usuario, validación, eventos y conversión en el servidor, centraliza la navegación de las páginas web.

La especificación de JSF fue desarrollada por la Java CommunityProcess como JSR 127, que definía JSF 1.0 y 1.1, JSR 252 que define JSF 1.2 y JSR 314 para JSF 2.0.

Estos objetivos de diseño representan el foco de desarrollo de JSF:

- Definir un conjunto simple de clases base de Java para componentes de la interfaz de usuario, estado de los componentes y eventos de entrada. Estas clases tratarán los aspectos del ciclo de vida de la interfaz de usuario, controlando el estado de un componente durante el ciclo de vida de su página.
- Proporcionar un conjunto de componentes para la interfaz de usuario, incluyendo los elementos estándares de HTML para representar un formulario. Estos componentes se obtendrán de un conjunto básico de clases base que se pueden utilizar para definir componentes nuevos.
- Proporcionar un modelo de JavaBeans para enviar eventos desde los controles de la interfaz de usuario del lado del cliente a la aplicación del servidor.
- Definir APIs para la validación de entrada, incluyendo soporte para la validación en el lado del cliente.
- Especificar un modelo para la internacionalización y localización de la interfaz de usuario.
- Automatizar la generación de salidas apropiadas para el objetivo del cliente, teniendo en cuenta todos los datos de configuración disponibles del cliente, como versión del navegador.

Fases del JSF

Ciclo de vida de JSF

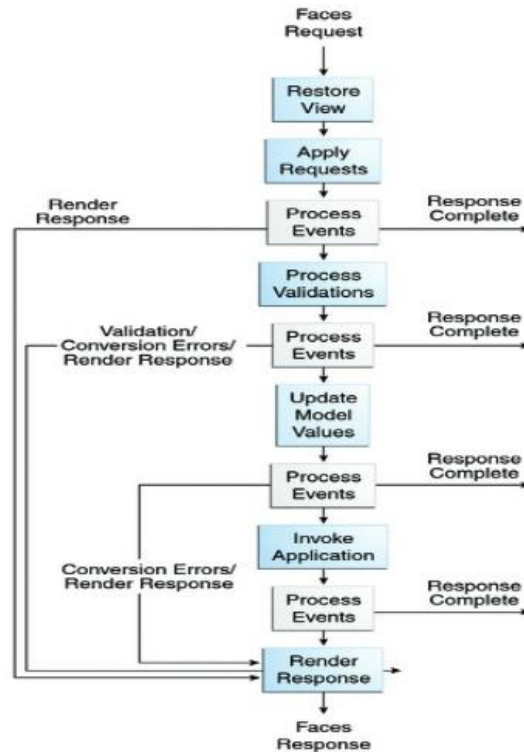


Figura 3. Muestra las fases que cumple desde el momento de hacer clic

Fuente: (docs.oracle.com, 2013).

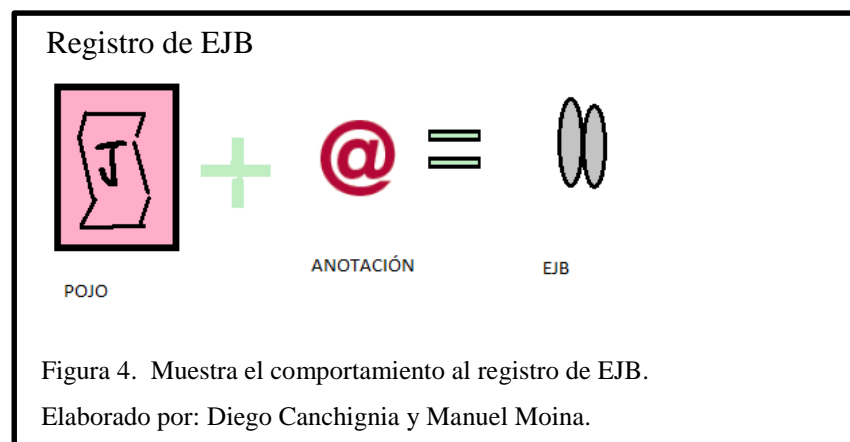
- **Fase de Recuperación (Restore View):** En esta fase JSF construye la vista de la página, conectando validadores con los componentes correspondientes, almacena el resultado en FacesContext, donde se tiene los requerimientos e información necesaria para procesar la petición.
- **Fase de aplicación de los valores de la petición (Apply Request Values):** A partir de los parámetros de las peticiones y restaurado el árbol de componentes cada uno obtiene un nuevo valor. Si ocurriese un problema en dicho proceso se genera un mensaje de error que se pone en una cola dentro del contexto y se mostrará en la fase de renderizado, junto a los generados en la fase de validación.
- **Fase de Validación (Process Validations):** En esta fase se ejecuta la validación de los valores de los componentes.
- Si el valor no es válido, se despliega un mensaje de error al contexto que se enruta hasta la fase de renderizado.

- **Fase de Actualización de Valores (Update Model Values):** Si los valores son válidos, se sincronizan las propiedades del objeto en la parte de los servers con los valores del componente.
- **Fase de Invocación de la Aplicación (Invoke Application):** JSF maneja eventos en el nivel de aplicación, puede enviar formularios a otras páginas.
- **Fase de renderización de la Respuesta (Render Response):** JSF envía el renderizado al contenedor JSP si la aplicación lo ocupase. Si la petición es inicial, los componentes en la página se cargarán en el árbol de componentes conforme JSF ejecuta la página. Si surgieran errores, se mostrarán en la página. Una vez renderizada la vista, la respuesta se guarda, para ser llamada.

1.3.2.1.2. Enterprise Java Bean (EJB)

Simplifican procesos de desarrollo en JAVA de aplicaciones de componentes empresariales distribuidos. Están en el lado del server y ejecutan la lógica de negocio y son llamadas desde host locales y remotos.

Para simplificar el desarrollo adopta un modelo de programación Plain Old Java Object (POJO). El registro de EJB se realiza mediante una anotación agregada a POJO como se muestra en la figura 4.



POJO puede interpretarse como “Un objeto Java Plano Antiguo”. POJO es una instancia simple de una clase (no se extiende, no se implementa), dependen de un framework en especial. POJO es un objeto ligero.

EJB es parte de la plataforma JavaEE, desarrollada y mantenida por Sun Microsystems, la misma que en la actualidad conforma parte de Oracle Corporation.

EJB apartir de 3.0 no está ligado a ninguna obligación o restricción a objetos de negocio algún API en concreto, por lo que se puede reutilizar componentes, es un adelanto de la programación orientada a objetos. Es considerado como un contenedor donde se alojan los componentes.

Servicios que EJB proporciona

Los más importantes son los siguientes:

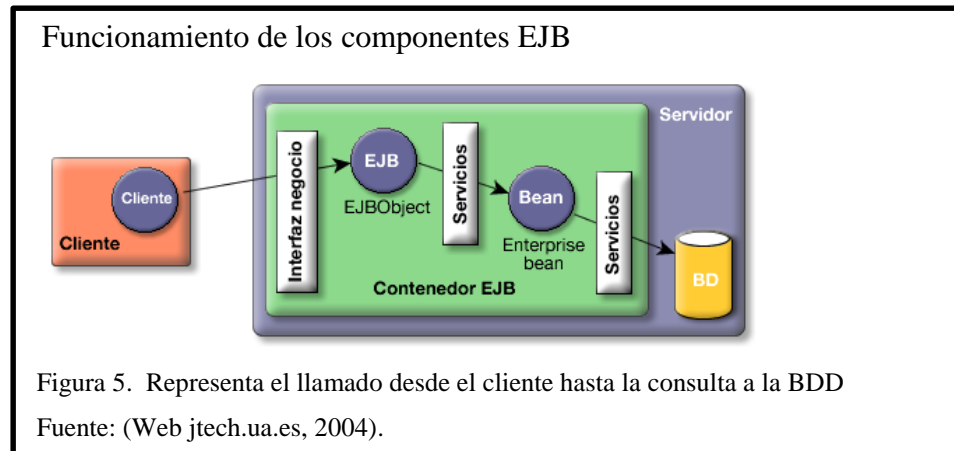
- Manejo de transacciones: Abre y cierra transacciones de los métodos del bean a los que son llamados
- Seguridad: Comprueba los permisos de acceso a los métodos del bean
- Concurrencia: De varios clientes pueden ser llamados al mismo tiempo
- Servicios de red: Desde host distintos se comunican el cliente y el método
- Gestión de recursos: Gestiona recursos múltiples, como fuentes de datos heredadas de distintas aplicaciones, el acople de distintos lenguajes
- Persistencia: Sincronización de datos en los métodos beans y una base de datos
- Escalabilidad: Creación de clusters de servers de aplicaciones con múltiples hosts para controlar el incremento de carga
- Adaptación en tiempo de despliegue: Modificaciones de características del despliegue del bean

Funcionamiento de los componentes EJB

Se basa del trabajo del contenedor EJB que es un programa Java que se ejecuta en el server el cual contiene las clases y objetos para el funcionamiento de los Beans.

Al realizar peticiones a los beans, el servidor que contiene el bean que se encuentra ejecutándose en máquinas virtuales Java distintas, o, pueden estar en distintos hosts. El cliente no se comunica directamente con el enterprise bean, por esta razón el contenedor EJB proporciona un objeto EJB que hace de interfaz para todas peticiones de los clientes

o llamado al método de negocio del enterprise bean, este objeto EJB solicita al contenedor EJB una serie de servicios y se comunica con el enterprise bean. Por último, el bean realiza las peticiones a la base de datos. Para tener un poco claro se representará en la figura 5.



Tipos de EJB

EJB de sesión (Session EJB): Proceso de negocio que será ejecutado de forma sincrónica, y gestiona en el server el flujo de la información. Se puede describir dos tipos que son los siguientes:

- **Con Estado (Stateful).**- Son objetos distribuidos con un estado, no es persistente, garantiza el mantenimiento del estado de los componentes para varias transacciones, pero es limitado para un solo cliente para el acceso.
- **Sin Estado (Stateless).**- Son objetos distribuidos con un estado asociado, acceso concurrente, son más ligeros que los stateful, pero es limitado ya que no garantiza que el EJB guarde su estado.

EJB dirigidos por mensajes (Message-driven EJB): Proceso de negocio que será ejecutado de forma asincrónica, utiliza Java Messaging System (JMS). El cliente no debe instanciarlo.

EJB de entidad (Entity EJB): Representan un objeto en alguna base de datos de la empresa, es una instancia que se representa en una fila en una tabla de la base de datos.

1.3.2.1.3. Java Persistence API (JPA)

JPA Provee de soporte a persistencia de objetos, define un mapeo de objeto a relación mediante anotaciones o ficheros XML⁴. Gestiona las consultas, cargas, escrituras y modificaciones de las entidades, entidad es un objeto java que muestra datos almacenados en una lista con un número limitado de objetos de una base de datos, esto significa que es una clase JAVA con metadatos que describe una tabla de una base de datos, estos metadatos pueden ser anotaciones o XML.

JPA con sus implementaciones permite que las aplicaciones sean portables.

JPA consta de tres partes:

- API de Java Persistence (javax.persistence.package)
- Consultas de Java Persistencees Query Language (JPQL)
- Metadatos objeto/relación

Dependiendo de dónde se utilicen las anotaciones las entidades puedes ocupar campos o propiedad persistentes. Las propiedades o campos pueden estar con los siguientes tipos de datos:

- Tipo String
- Tipo numérico
- Tipo primitivo de JAVA
- Objeto serializable (BigInteger, BigDecimal, etc)

JPA se clasifica en las siguientes anotaciones:

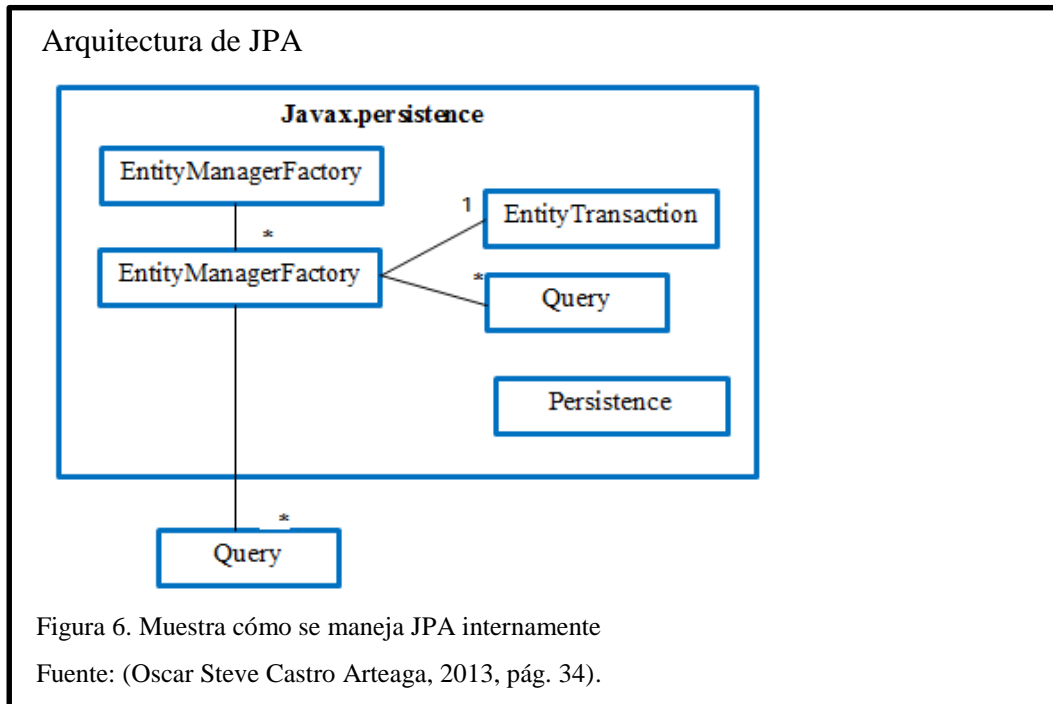
Mapeo Lógico: Describe asociaciones de clases.

- @OneToOne
- @OneToMany

4. XML (eXtensible Markup Language) o Lenguaje de Marcas Extensible es estándar de datos que puede codificar el contenido

Mapeo Físico: Describe los esquemas físicos de las bases de datos.

- @Column
- @JoinColumn



1.3.2.2. Primefaces

PrimeFaces es un componente para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2 es un Framework Open Source desarrollada por Prime Technology. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes como por ejemplo RichFaces⁵.

PrimeFaces cuenta con varias etiquetas que son de un uso fácil y de componentes de alta funcionalidad.

Propiedades:

- Conjunto de componentes (Editor de HTML⁶, autocompletar, cartas, gráficas o paneles, entre otros)
- Soporte de ajax⁷ con despliegue parcial, lo que permite controlar cuáles componentes de la página actual se actualizarán y cuáles no
- 25 temas prediseñados visuales y herramientas de edición
- Utiliza JQuery⁸ de framework JavaScript
- Cuenta con versiones para dispositivos móviles

1.3.2.2.1. Justificativo de utilizar este Framework

Para aclarar el justificativo del uso de Primefaces se muestra en la siguiente tabla las características del Framework:

5. **RichFaces**. Componente para JSF de código abierto para aplicaciones web.

6. **HTML**. HyperText Markup Language es un lenguaje de programación que se utiliza para el desarrollo de páginas web.

7. **AJAX**. Asynchronous JavaScript And XML técnica de desarrollo web para crear aplicaciones interactivas.

8. **JQuery**. Simplifica la sintaxis al escribir un código de desarrollo.

Tabla 2.

Comparación de Frameworks

Característica	ICEfaces	RichFaces	Primefaces
Soporte Ajax	Mediante la propiedad <code>partialSubmit</code> ⁹ se implementa de forma nativa	Utiliza Ajax4JSF ¹⁰ con componentes no visuales, además de introducir los componentes de RichFaces,	Hay que indicarlo explícitamente, por medio de atributos específicos de cada componente
Librerías en las que se basan	Usa el soporte de <code>prototypej</code> ¹¹ , utiliza <code>script.aculo.us</code> .	Usa el soporte de <code>prototypejs</code> y <code>script.aculo.us</code> , también <code>jquery</code> .	Utiliza el soporte de <code>jQuery</code> y <code>jQuery UI</code> para los efectos visuales.
Personalización de la interfaz de usuario	Incorpora skins ¹² y distribuye 3 temas.	Incorpora skins y distribuye 12 temas.	Incorpora skins, utiliza ThemeRoller ¹³ , y dispone de 35 temas prediseñados.
Número de componentes	Tiene 79 componentes en la versión básica, 32 de la versión empresarial,	Tiene 212 componentes entre RichFaces y Ajax4JSF. RichFaces tiene los componentes son OpenSource y podemos usar un Pick List sin contratar nada, sin embargo, con ICEfaces sin queremos un Dual List o pagamos o lo implementamos nosotros.	Tiene más de 90 componentes OpenSource, como el HTMLEditor, un kit para crear interfaces web para teléfonos móviles.
Licencia	MPL 1.1 ¹⁴ , que cubre la LGPL V 2.1. Si bien disponen de una versión empresarial con licencia comercial.	LGPL ¹⁵ V 2.1. en su totalidad.	Apache License V2
Relevancia	Ha sustituido a Woodstock como librería de componentes de Sun para el desarrollo de aplicaciones RIA ¹⁶ . Se distribuye, por defecto, con NetBeans.	Es la librería de componentes visuales de Jboss, se integra, por defecto con Jboss Seam, aunque éste también soporta ICEfaces.	Ha sido una de las primeras librerías capaces de integrarse con JSF 2 y viene pisando fuerte debido a la diversidad y calidad de sus componentes. Puede utilizarse junto a Richfaces, pero no es compatible con ICEfaces.

Nota: Tener una idea clara del porque ocupar estos frameworks

Fuente: (Enrique Viñé Lerma, 2010, pág. 19).

9. Partialsubmit. Permite refrescar solo componentes más no la página web completa.

10. Ajax4JSF Componente para JSF de código abierto para aplicaciones web.

11. Prototypej. Prototipo de java para elimina la complejidad de la programación web del lado del cliente.

12. Skins. Apariencia diferente para un aplicativo.

13. ThemeRoller. Aplicación web con interfaces intuitivas para diseño de páginas con temas para descarga.

14. MPL 1.1. Mozilla Public License (licencia pública de Mozilla) es una licencia de software libre.

15. LGPL Lesser General Public License (Licencia Pública General Menor) garantiza a los usuarios finales la libertad de usar, estudiar, copiar y modificar el software.

16. RIA. Rich Internet Application (Aplicaciones de Internet Enriquecidas) es una aplicación web combinada con las ventajas que ofrece las aplicaciones web y las aplicaciones de escritorio.

1.3.2.3. GlassFish

Es un servidor de aplicaciones basado en software libre, que implementa tecnologías que están definidas en la plataforma de Java Enterprise Edition (JEE), que a su vez puede ejecutar otras aplicaciones que cumplan las especificaciones. Está basado en un código fuente de SUN Microsystems y Oracle Corporation, la primera empresa desarrollo la aplicación y la segunda la adquirió. Cuenta con dos versiones la free es GlassFish Enterprise Server y la versión comercial es Oracle GlassFish Enterprise Server.

GlassFish es un derivado de Apache Tomcat y lo utiliza como contenedor de Servlets, Tomcat es también un servidor de aplicaciones y usan un componente adicional el cual es Grizzly que incorpora Java NIO para mejorar y dar escalabilidad y velocidad.

GlassFish reconocido por un ícono peculiar que es un pez, y tiene las siguientes ventajas:

- Preparado para ejecutarse en entornos clustering con failover.
- Ruta de migración sencilla.
- Administración y supervisión sencilla y fácil de manipular.
- Compatible con lenguajes de scripts.

La administración con su IDE se lo puede manejar por JDeveloper, IntelliJ, Eclipse y Netbeans, lo que nos permite administrarlo desde algún browser para navegación web.

1.3.2.3.1 Justificativo de utilizar GlassFish

Una ventaja que tiene GlassFish sobre Tomcat es la capacidad de retener sesiones entre distintos despliegues de aplicaciones, ahorro de tiempo para desarrollo.

Otra ventaja es que facilita la reconfiguración dinámica de servidores virtuales.

Tabla 3.

Comparación entre GlassFish y Tomcat

	GLASSFISH (ms)	TOMCAT (ms)
Operaciones/segundo	6988,9	6615,3
Tiempo medio respuesta	0,242	0,358
Tiempo máx. respuesta	1,519	3,693
90% tiempo respuesta	0,6	0,75

Nota: Muestra las operaciones sobre segundo de los distintos contenedores para incrementar el número de usuarios con 16,000 usuarios

Fuente: (Web Blogspot, 2010).

Al escoger un servidor de aplicaciones hay que tener precaución en la carga y estrategia de negocio, algunas operaciones de la institución. Sun afirma que GlassFish ha mostrado superioridad sobre Tomcat para desarrolladores de nuevas generaciones.

1.3.2.4. MySQL

Es un motor de base de datos relacional, para soportar una gran carga de manera eficiente está su diseño de multihilo, y posee un amplio sublenguaje Structure Query Language (SQL). Fue creada por MySQL AB empresa sueca, cuenta con una versión libre y una comercial, la comercial cuenta con soporte técnico certificado y garantizado, se puede integrar con software propietario.

MySQL se considera en marketing como el gestor más utiliza en el software libre, por la rapidez y facilidad de uso. Cuenta con una infinidad de librerías y herramientas que le vuelve compatible convarios lenguajes de programación, a eso se suma la fácil instalación y configuración.

1.3.2.4.1. Características de MySQL

Inicialmente Mysql carecía de integridad referencial y transacciones, lo cual fue corregido y potenciado, por lo que tenemos las principales características que son las siguientes:

- Velocidad y Robustez
- Aprovecha la potencia de sistemas multiprocesador
- Implementación multihilo
- Soporta alta cantidad de tipos de datos para las columnas
- Cuenta con API's para diferentes lenguajes (C, C++, Java, PHP, etc)
- Alta portabilidad entre sistemas
- Soporta hasta 32 índices por tabla
- Administración de usuarios y passwords con alto nivel de seguridad en los datos
- Requerimiento de bajo costo para elaborar bases de datos
- Fácil instalación y configuración
- Alta conectividad y seguridad
- Probabilidad baja de corromper datos

CAPÍTULO 2

ANÁLISIS Y DISEÑO DEL SISTEMA INFORMÁTICO

Hospital de Especialidades Eugenio Espejo

Historia

Solamente existían hospitales en Quito, Guayaquil, Cuenca, Loja, Riobamba, Manabí y Babahoyo; administrados por civiles y en su gran mayoría la atención no la realizaban médicos sino barchilones es decir practicantes. No existía una Escuela de Enfermería bien preparada ni organizada, siendo las Hermanas de la Caridad las que cumplían esta función (Página web oficial del Hospital Eugenio Espejo, 2013).

Desde fines del Siglo pasado ya se comenzó a gestar en Quito la idea de construir una nueva Casa de Salud. Las primeras ideas ya concretas de construir un nuevo hospital que reemplace al antiguo San Juan de Dios aparecen hacia 1890. En ese entonces el San Juan de Dios se encontraba deteriorado, desprovisto de ciertos adelantos de infraestructura hospitalaria para la época y ya no daba cabida adecuada y suficiente a la demanda de pacientes. Se procedió a solicitar a la Facultad de Medicina que presente un proyecto acerca de la nueva edificación (Página web oficial del Hospital Eugenio Espejo, 2013).

La Facultad delegó al Doctor Manuel María Casares quien fue decano de la misma para que realice esta labor, él presentó una comparación entre los hospitales franceses y norteamericanos, las ventajas y desventajas de cada uno, además señaló ciertas modificaciones o adecuaciones que se requerían para tal obra, basándose en los Hospitales Montpellier, señaló los detalles de las instalaciones de agua, canalización y la adquisición de un sifón hidráulico que aún no se conocía en Quito, además por la falta de canalización en la ciudad, él sugería la creación de letrinas especiales. Indicó la forma de construir la sala de operaciones y el uso de la estufa de Poupinel (Página web oficial del Hospital Eugenio Espejo, 2013).

Lamentablemente el proyecto del Doctor Casares no fue aceptado por el Presidente Luis Cordero, especialmente en lo que significa la ubicación del mismo: al norte la calle Salinas, al Oeste la Plaza de la Recoleta y al sur-este el río Machángara, por este motivo, Cordero nombra a los doctores Miguel Egas y Rafael Arjona Silva para que proponga otro lugar donde edificar este hospital. Los doctores Egas y Arjona proponen que el nuevo hospital se lo construya al sur-oeste del Itchimbía y que el agua que no existía en el mencionado sitio, se la traiga de La Alameda o de la Plaza de San Blas. Hubo gente que aun sin siquiera conocer el sitio propuesto se oponía, de todos modos primó más la razón y se aprobó construir el nuevo hospital en sitio propuesto por los doctores Egas y Arjona Silva (Página web oficial del Hospital Eugenio Espejo, 2013).

En 1901, la Junta de Beneficencia de Quito vuelve a interesarse por la construcción de esta Casa de Salud y emprende una campaña altruista para esta obra, consigue apoyo de algunos filántropos de indiscutido interés por la causa social como Mariano Aguilera, Alejandro Mosquera, Abel Guarderas, la Municipalidad Quiteña y otros. Y es así que un memorable 23 de mayo de 1901 la salud en el Ecuador inicia un nuevo derrotero al colocarse la primera piedra para la construcción del nuevo hospital. Sin embargo se coloca la primera piedra sin existir el terreno debidamente concedido para la obra, ni el presupuesto del que sería el Hospital Eugenio Espejo, en los dos meses subsiguientes se adquiere el terreno a los señores Benjamín Piedra y Francisco Andrade gracias a la influencia e interés de la Junta de Beneficencia (Página web oficial del Hospital Eugenio Espejo, 2013).

Es incuestionable el interés del presidente Eloy Alfaro por mejorar las paupérrimas escalas de salud y educación que existían en nuestro país. Una de las frases que se hicieron populares en la Alfarada fue “hospitales para los pobres, asilos para los ancianos”; Alfaro fue el presidente que más hizo por la educación y la salud, sus enunciados no se quedaron solamente en el papel, a pesar de la oposición que tuvo y las trabas que enfrentó, el régimen alfarista durante sus varios períodos es el que más construyó y aportó con escuelas, colegios, colegios técnicos, normales y militares, hospitales, maternidades, etc, continuando lo iniciado por García Moreno, aunque debe

quedar claro que Alfaro cualitativa y cuantitativamente fue mejor, incluyendo el hecho de no contar con el apoyo de la Iglesia y de su líder máximo González Suárez, quien ha quedado para la historia como un sacerdote de ambivalente actuación (Página web oficial del Hospital Eugenio Espejo, 2013).

El Presidente Alfaro en 1908 en su mensaje a la nación se refirió profundamente conmovido sobre la pobreza de las instituciones hospitalarias y la necesidad imperiosa de que se disponga fondos para su funcionamiento, propuso que se utilice los fondos de los bienes llamados de Manos Muertas para ayudar al desvalido y a la beneficencia. La Cooperativa de José Peralta en todos estos empeños fue de gran eficacia (Página web oficial del Hospital Eugenio Espejo, 2013).

Sin embargo el inicio de la construcción del nuevo hospital no se plasmó en hechos hasta el año de 1911. Como había indicado inicialmente el Hospital San Juan de Dios ya no daba abasto a la demanda actual y por lo tanto la creación de un nuevo hospital se volvía imperiosa, por lo que ese año y finalizando la última Presidencia Constitucional del General Eloy Alfaro Delgado (1 de enero de 1907 – 11 de agosto de 1911) se autoriza a que la Junta de Beneficencia venda ciertos bienes de su propiedad para que el producto de esta venta se lo utilice en la construcción de un nuevo hospital y maternidad. En el año de 1912, a la edificación del hospital que se llamará Eugenio Espejo se la declara como obra nacional de parte del Gobierno (Página web oficial del Hospital Eugenio Espejo, 2013).

El nuevo hospital edificado sobre dos plantas estaría conformada por varias salas o pabellones (seis pabellones) separados entre sí, como eran al principio del siglo los hospitales franceses y norteamericanos. Los pabellones se dividirían en los que se dedicaban a la atención de clínicas y a cirugía (Página web oficial del Hospital Eugenio Espejo, 2013).

Es así que en 1933, en el gobierno del liberal Juan de Dios Martínez Mera se inaugura el Hospital Policlínico Eugenio Espejo con una capacidad máxima de internamiento de

quinientas camas. Luego de aproximadamente cuatrocientos años de depender la salud de los quiteños del recordado y venerable Hospital San Juan de Dios la capital del Ecuador cuenta con un hospital de arquitectura horizontal y tecnología que se prestó para el buen uso de las ciencias médicas de la Universidad Central que tuvo un espacio moderno donde ejercer la docencia, formando a los nuevos galenos (Página web oficial del Hospital Eugenio Espejo, 2013).

Misión

Prestar servicios de salud con calidad y calidez en el ámbito de asistencia especializada, a través de su cartera de servicios, cumpliendo con la responsabilidad de promoción, prevención, recuperación, rehabilitación de la salud integral, docencia e investigación, conforme a las políticas del Ministerio de Salud Pública y el trabajo en red, en el marco de la justicia y equidad social (Página web oficial del Hospital Eugenio Espejo, 2013).

Visión

Ser reconocidos por la ciudadanía como un hospital accesible, que presta una atención de calidad que satisface las necesidades y expectativas de la población bajo principios fundamentales de la salud pública y bioética utilizando la tecnología y los recursos públicos de forma eficiente y transparente (Página web oficial del Hospital Eugenio Espejo, 2013).

Quirófanos del Hospital de Especialidades Eugenio Espejo

La institución cuenta con un área de quirófanos para cirugías quirúrgicas programadas, son 10 quirófanos en el área de Quirófano central en los que se utilizan diariamente para intervenciones.

Un quirófano es una habitación que se encuentra en hospitales o centros de atención médica y son acondicionadas especialmente para práctica de cirugías quirúrgicas a

pacientes que así lo demanden. Además, en el quirófano, se ejercen otras actividades relacionadas, como: el suministro de anestesia, una acción de reanimación, entre otras.

En el Hospital de Especialidades Eugenio Espejo por estar catalogado como un Hospital de Tercer Nivel cuenta con 36 especialidades para consulta externa y tratamientos, pero solo son 18 especialidades que ingresan a quirófanos para cirugías (Estadística de pacientes mensuales HEE 2014, 2014).

En los quirófanos de la institución intervienen aproximadamente 1000 cirugías quirúrgicas mensuales.

2.1. Especificación de requerimiento

Dentro del Hospital de Especialidades Eugenio Espejo se ha detectado la necesidad de automatizar los siguientes requerimientos: La gestión del proceso del parte operatorio (ver anexo 3, parte operatorio), registro de la solicitud de cirugía, registro de la información de los resultados de la cirugía, con esto se logra optimizar los tiempos y facilitar el acceso a los datos.

El proceso general para ingreso a cirugía.

Subproceso 1: El paciente ingresa por consulta externa, para chequeo de un especialista por alguna enfermedad, activa subproceso dos.

Subproceso 2: El especialista realiza un diagnóstico del paciente, decide si necesita cirugía o no.

Sí, ingreso de paciente a hospitalización, activa subproceso 3.

No, envía medicación y nueva consulta.

Subproceso 3: El departamento de estadística procede al registro de datos, y notifican en hospitalización el ingreso a piso del paciente, dependiendo de la especialidad, activa subproceso 4.

Subproceso 4: Una vez hospitalizado es nuevamente valorado por especialistas, activa subproceso 5.

Subproceso 5: El paciente es sometido a exámenes médicos, solicitados en el subproceso 4, y decide si el paciente intervenido quirúrgicamente:

Sí, el paciente ingresa al proceso de tratamiento posquirúrgico, activa subproceso 8.

No, verifica si cumple con los requisitos pre-quirúrgicos y decide lo siguiente:

Sí, se registra en el parte operatorio la solicitud, activa subproceso 6.

No, el paciente ingresa al proceso de tratamiento pre-quirúrgico hasta estabilizarlo y poder planificar el parte operatorio, regresa al subproceso 4.

Subproceso 6: Se verifica si necesita medicamentos y/o implementos médicos, se decide lo siguiente:

Sí, descargo de medicamentos y/o insumos médicos, se procede a la cirugía.

No, Se procede a la cirugía con implementos básicos.

Subproceso 7: Terminada la cirugía, se registra en el parte operatorio, la información del procedimiento y observaciones, si fue satisfactorio o no:

Sí, paciente a recuperación, regresa a subproceso 4.

No, cadáver a la morgue

Subproceso 8: El paciente sale del hospital con el alta médica.

Este proceso se encuentra ilustrado en la figura 7 que se encuentra a continuación.

Diagrama de proceso

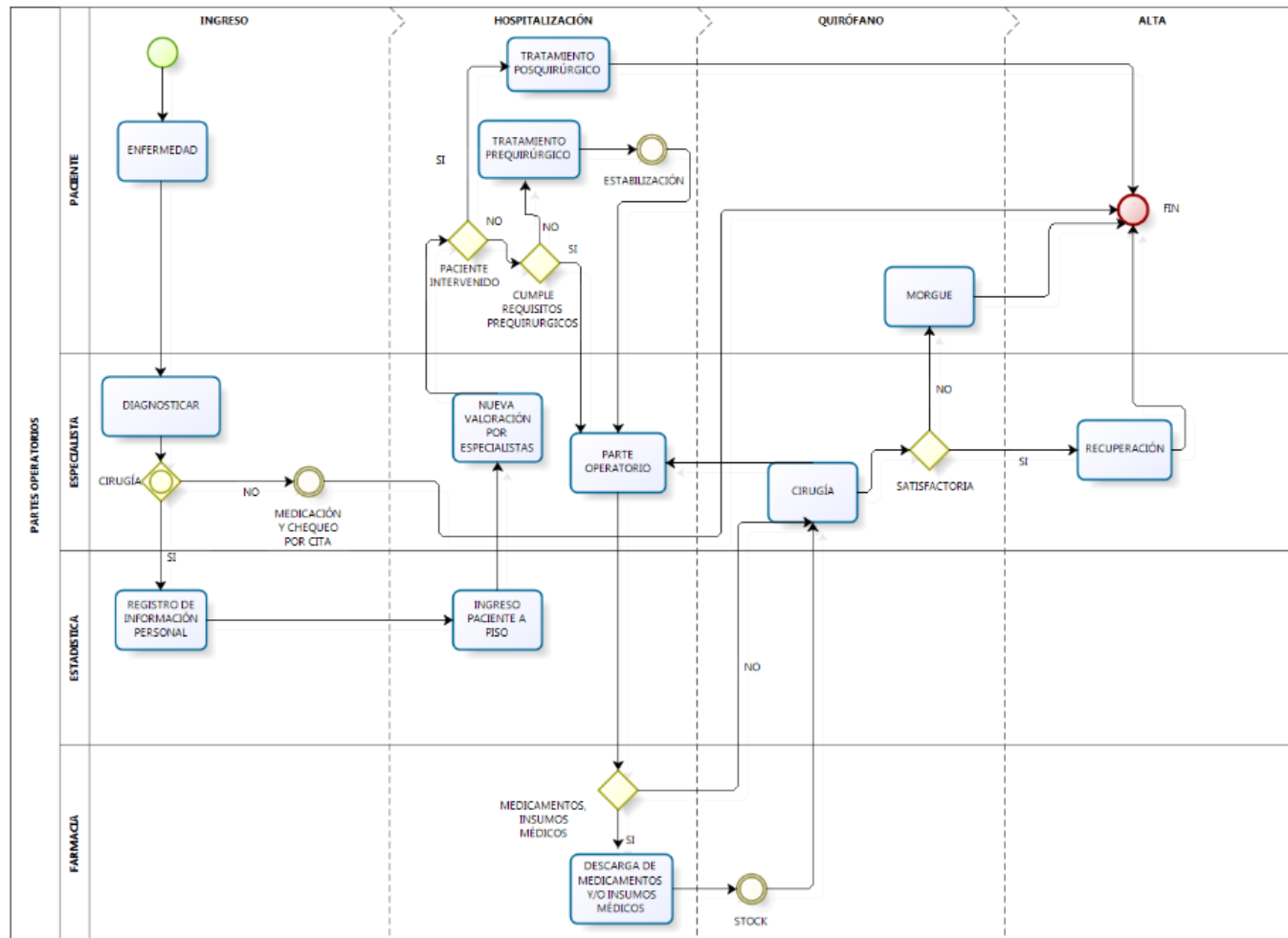


Figura 7. Para la solicitud y registro de un parte operatorio

Elaborado por: Diego Canchignia y Manuel Moina.

2.1.1. Recopilación de información para la pila de producto

Mediante una entrevista, se recopila información necesaria para el desarrollo del proyecto, para lo cual se realiza una serie de encuestas.

Las entrevistas están realizadas a personas que se encuentran directamente relacionadas al proceso del parte operatorio entre los cuales están:

- Secretaría de Especialidad
- Secretaría de Quirófano
- Secretaría de Estadística
- Especialistas

2.1.1.1. Historia de usuario

Tabla 4.

Historia de usuario 1

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador de la Aplicación
Nombre Historia: Creación, modificación y eliminación de usuarios, roles	
Prioridad en Negocio: Baja	
Puntos Estimados: 2	Iteración Asignada: 1
Programador Responsable: Manuel Moina	
Descripción: La aplicación web tendrá roles según las funciones, por ende se necesita usuarios y contraseñas que permita el ingreso al sitio web, y según el rol a las diferentes opciones.	
Como probarlo: <ul style="list-style-type: none">➤ Ingresar a la aplicación web con el usuario y contraseña➤ Revisar el menú que puede acceder según el rol	

Nota: Muestra al usuario que administrará la aplicación para crear usuarios y perfiles

Elaborado por: Diego Canchignia y Manuel Moina.

Tabla 5.

Historia de usuario 2

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador de la Aplicación
Nombre Historia: Registro de profesionales y/o especialistas	
Prioridad en Negocio: Media	
Puntos Estimados: 3	Iteración Asignada: 1
Programador Responsable: Manuel Moina	
Descripción: Permitirá el registro, modificación y deshabilitación de profesionales y/o especialistas en la aplicación web, estos pueden ser médicos especialistas, anestesiólogos, enfermeras, y ayudantes.	
Como probarlo: <ul style="list-style-type: none"> ➤ Ingresar a la aplicación web con el usuario y contraseña ➤ Revisar el menú, ingresar a la opción para registro de profesionales ➤ Registrar un profesional 	

Nota: Muestra al usuario que administrará la aplicación para el registro de profesionales

Elaborado por: Diego Canchignia y Manuel Moina.

2.1.1.2. Product Backlog (Pila de producto)

Los requerimientos del Hospital de Especialidades Eugenio Espejo se encuentran registrados dentro de la siguiente tabla de pila de productos:

Tabla 6.

Pila de producto

Pila de producto - Análisis, Diseño e Implementación de una Aplicación Web para la administración y gestión de partes operatorios en quirófanos del Hospital de Especialidades Eugenio Espejo						
ID	NOMBRE	DESCRIPCIÓN	IMPORTANCIA (1 - 150)	ESTIMACIÓN INICIAL (DÍAS)	COMO PROBARLO	NOTA
1	Control de acceso	Para acceder a la aplicación es necesario contar con una cuenta de usuario.	120	5	A través del formulario de autenticación registrar credenciales (usuario y clave) de usuarios, luego ingresar a la página de autenticación de usuario y digitar las credenciales para comprobar el acceso o restricción a la aplicación.	Es necesario crear un módulo de administración de usuarios, el cual nos permita crear, editar y bloquear usuarios. Solo el Administrador de la aplicación puede manejar el módulo de administración de usuarios, para mayor seguridad las claves deben estar encriptadas.
2	Definición de roles	Un usuario puede ingresar a la aplicación y acceder solo a la información a la que esta autorizado	100	5	Crear usuarios con diferentes roles, registrarse en la aplicación y verificar que solo puedan acceder a la información que esta definida para su rol.	Es necesario definir los roles que va a tener la aplicación, además se debe permitir administrar los permisos que tiene cada rol de usuario, los mismos que podrán cambiar es decir dar mas permisos o quitar permisos por rol de usuario. Solo el administrador de la aplicación puede crear, editar, bloquear, dar o quitar permisos a los roles creados.
3	Registrar profesionales	Permitir registrar información de los profesionales (Especialistas, Anestesiólogos, Enfermeras) en la aplicación	110	8	Mediante el formulario de registro de especialistas registrar información personal y profesional de los especialistas según el cargo que desempeña en el Hospital de Especialidades Eugenio Espejo, luego verificar que el registro se guardo o modifiko según el caso en el listado de especialistas registrados en la aplicación.	La información de los profesionales será tomada del sistema de agendamiento de turno en consulta externa (Shaman ¹⁸), la información que no se encuentre registrada deberá registrarse en el sistema. De acuerdo al perfil del usuario este podrá modificar la información registrada.
4	Registrar parte operatorio	facilitar el ingreso y edición de la información que conforma un parte operatorio.	140	10	En el formulario de registro de parte operatorio ingresar la información que contiene un parte operatorio restringiendo fechas de acuerdo al calendario de atención del médico.	La aplicación debe facilitar el ingreso de información correcta en el parte operatorio, mediante el número de historia clínica se cargara la información del paciente necesaria para llenar el parte operatorio sin embargo la aplicación debe permitir editar la información cargada automáticamente por la aplicación.
5	Publicación de programación de cirugías.	Una vez ingresada y aprobada la cirugía esta deberá publicarse en la aplicación.	100	5	Verificar mediante la programación de cirugías aprobadas las publicaciones que se muestran.	Las publicaciones deben cumplir con el reglamento y políticas establecidas por la LOTAIP ¹⁹ .
6	Informes	Permitir consultar cirugías realizadas por médico o especialidad.	100	5	Ingresar a la aplicación y registrar un parte operatorio, ir a la página de reportes y realizar la búsqueda y comprobar que la cirugía registrada se encuentre en el resultado obtenido.	El resultado de la consulta debe mostrarse en tablas con paginación, para no distorsionar el formato de la página.

Nota: Registra los requerimientos para la aplicación web

Elaborado por: Diego Canchignia y Manuel Moina.

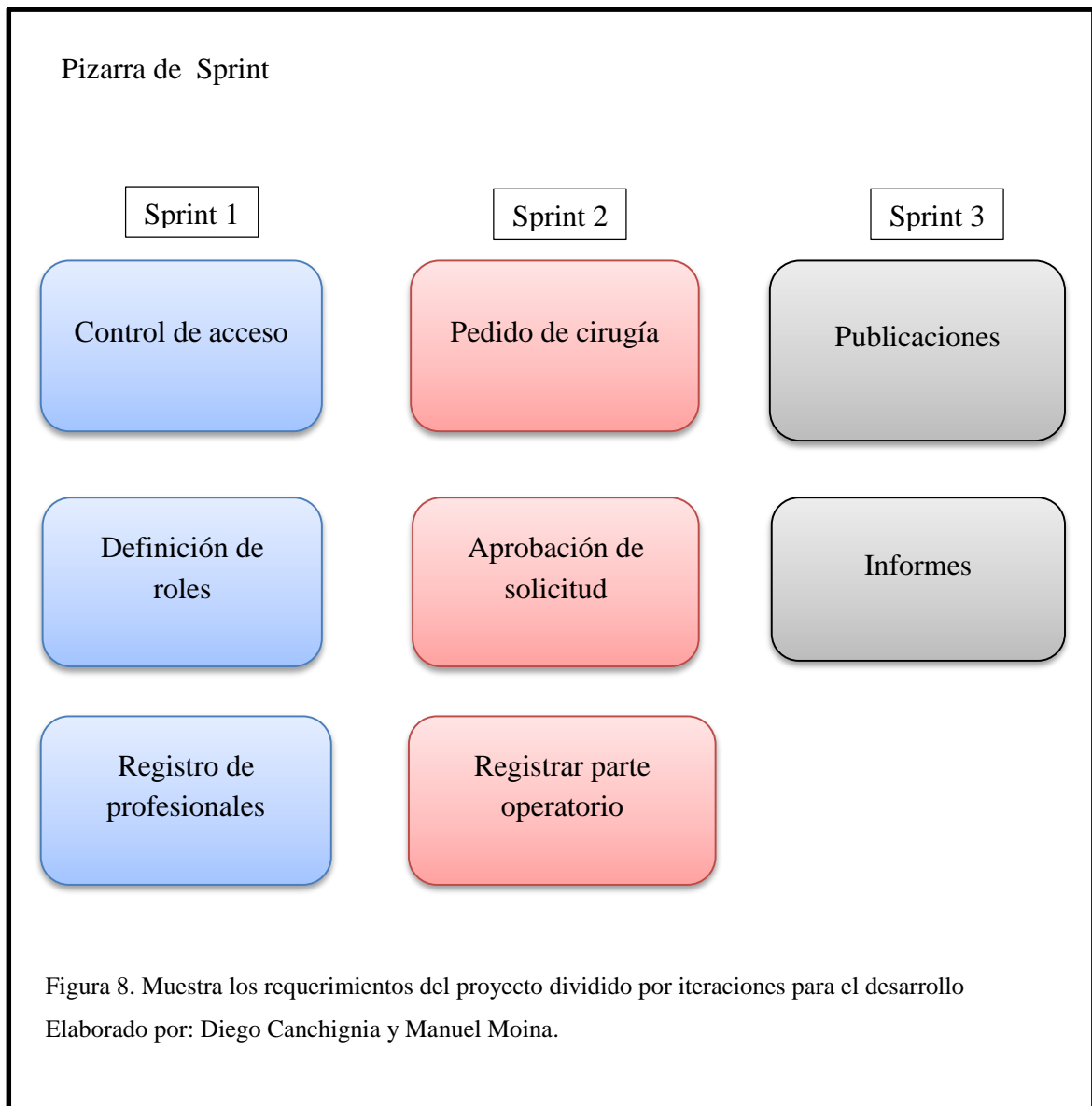
17. Shaman: Sistema de creación de números de historias clínicas, en la cual se ingresa datos personales de los pacientes (Nombre, teléfonos, dirección domiciliaria, información de contactos), y agendamiento de citas médicas con los especialistas del Hospital de Especialidades Eugenio Espejo. Este sistema cuenta con su base de datos, a la cual se realizan consultas para validar información del paciente (Molina Segovia, 2008)

18. LOTAIP. Ley Orgánica de Transparencia y Acceso a la Información Pública, garantiza el derecho para acceder a la información pública

2.2. Planificación del desarrollo

2.2.1. Sprint Planning (Planificación de las Iteraciones)

Los Sprints de la pizarra permiten visualizar de manera rápida las tareas que se ejecutarán en el desarrollo de la aplicación web.



2.2.1.1. Sprint 1

Al contar con la pizarra de Sprint, se plantea la elaboración del primer Sprint mediante una tabla, identificando las tareas a ejecutarse.

También van los días previstos para ejecutar cada tarea y el responsable de desarrollo.

Tabla 7.

Sprint 1

Sprint 1							
Fecha inicio	01/07/2014						
Fecha final	14/08/2014				N° Días Laborables	30	
					N° Días con prorroga	33	
N° tarea	Id Tarea	Descripción de tarea	Responsable	N° Días	N° Días Prorroga	Estado	Observaciones
1	T01	Control de acceso	Manuel Moina	10		En proceso	
2	T02	Definición de roles	Manuel Moina	10		En proceso	
3	T03	Registro de profesionales	Manuel Moina	10		En proceso	

Nota: Detalle de las tareas que se ejecutarán en el sprint

Elaborado por: Diego Canchignia y Manuel Moina.

2.2.1.1.1. Tarea 1 (Control de acceso)

Mediante una tabla se registra las observaciones en los avances diarios de la tarea, y si presenta algún inconveniente, se realiza una reunión para solventarlo.

Tabla 8.

Tarea 1 del control de acceso

T01		Control de Acceso			
N° Día	Avance Previsto	Avance Real	Observaciones	Solución	Estado tarea
1	10	10	Programación de Plantilla principal	Correcto	Terminada
2	10	10	Programación de casillero de usuario y password	Correcto	Terminada
3	10	10	Programación para olvido de contraseña	Correcto	Terminada
4	10	10	Programación de protección de claves	Correcto	Terminada
5	10	5	Campos en base de datos	Se agrega campo para profesionales	En proceso
6	10	1	Falta modificación en base de datos	Generar nueva base de datos con la corrección	En proceso
7	10	1	Falta modificación en base de datos	Generar nueva base de datos con la corrección	En proceso
8	10	1	Falta modificación en base de datos	Generar nueva base de datos con la corrección	En proceso
9	10	1	Falta modificación en base de datos	Generar nueva base de datos con la corrección	En proceso
10	10	0	Falta modificación en base de datos	Generar nueva base de datos con la corrección	En proceso
Total avance %	100	49			

Nota: Muestra la ejecución de la tarea y su inconveniente

Elaborado por: Diego Canchignia y Manuel Moina.

Al observar la tabla anterior, se compara mediante un gráfico estadístico los avances previstos con los avances realizados.

Tarea 1 del control de acceso

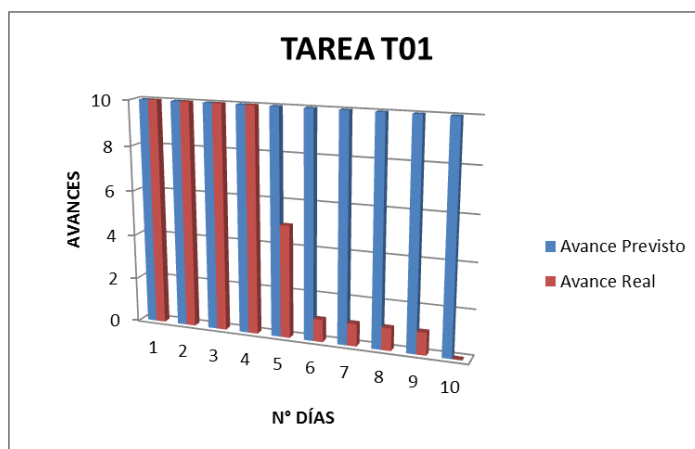


Figura 9. Muestra los avances diarios que se ejecutaron

Elaborado por: Diego Canchignia y Manuel Moina.

2.2.1.1.2. Inconveniente en la tarea 1 (Control de acceso)

Para mantener un registro de los problemas en la tarea, se detalla los inconvenientes a solventar, en esta tarea es la falta de una tabla que contenga roles como se puede observar en la siguiente figura.

Base de datos sin tabla para roles

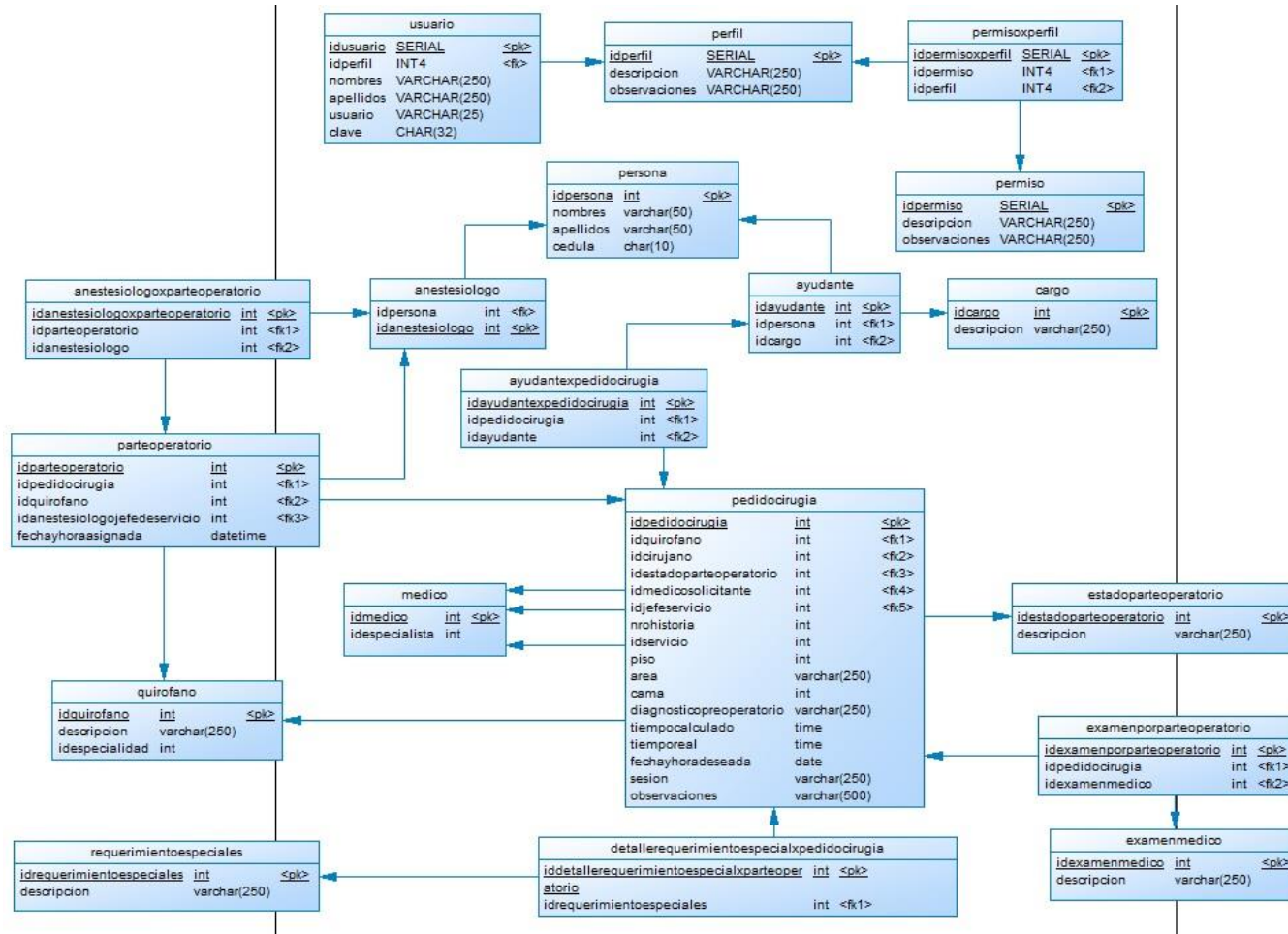


Figura 10. Muestra el diagrama de la base de datos con inconvenientes en la tarea 1 del control de acceso

Elaborado por: Diego Canchignia y Manuel Moína.

2.2.1.1.3. Tarea 1 terminada (Control de acceso)

Para contar con una bitácora de todo el proceso en el desarrollo de la tarea se registra cada acción efectuada.

Tabla 9.

Tarea 1 terminada del control de acceso

T01 Control de Acceso					
Nº Día	Avance Previsto	Avance Real	Observaciones	Solución	Estado tarea
1	10	10	Programación de Plantilla principal	Correcto	Terminada
2	10	10	Programación de casillero de usuario y password	Correcto	Terminada
3	10	10	Programación para olvido de contraseña	Correcto	Terminada
4	10	10	Programación de protección de claves	Correcto	Terminada
5	0	0	Se verifica factibilidad	Se realiza cambio	Corregido
6	10	10	Corrección de campo	Corrección de campo	Terminada
7	10	10	Insercción de información en la base de datos	Correcto	Terminada
8	10	10	Verificación de insercción en la base de datos	Correcto	Terminada
9	10	10	Programación del estilo de la plantilla	Correcto	Terminada
10	10	10	Programación del fondo de pantalla	Correcto	Terminada
11	10	10	Revisión de transición a siguiente tarea, siendo password correcto	Correcto	Terminada
Total avance %	100	100			

Nota: Muestra la tarea ejecutada al 100%, con el día incrementado bajo la petición en el Daily Scrum
Elaborado por: Diego Canchignia y Manuel Moina.

2.2.1.1.4. Daily Scrum (Control de Acceso)

Se registra lo que se tiene previsto desarrollarse en la tarea, de acuerdo a las conversaciones en las reuniones diarias, los inconvenientes encontrados, las soluciones, los avances, los participantes con sus responsabilidades.

Tabla 10.

Daily Scrum del control de acceso

T01	Control de Acceso			
Tiempo por día: 5 min				
Participantes	Nombre	Siglas	Responsabilidad	Scrum Team
	Diego Canchignia	DC	Scrum Master / Product Owner	
	Manuel Moina	MM	Desarrollador	
	Cliente	CHEE	Pruebas	
N° Día	Avance Previsto	Avance Real	Observaciones	Solución
1	10	10	Programación de Plantilla principal, revisado por DC, MM	Correcto
2	10	10	Programación de casillero de usuario y password, revisado por DC, MM	Correcto
3	10	10	Programación para olvido de contraseña, revisado por DC, MM	Correcto
4	10	10	Programación de protección de claves, revisado por DC, MM	Correcto
5	0	0	Falta campos en base de datos, se sugiere reunión con cliente, revisado DC,MM,CHEE	Se agrega campo para profesionales, y se solicita un día para solventar problema
6	10	10	Avance exitoso una vez corregido el inconveniente, revisado DC, MM	Correcto
7	10	10	Insercción de información en la base de datos, revisado por DC, MM	Correcto
8	10	10	Verificación de insercción en la base de datos, revisado por DC, MM	Correcto
9	10	10	Programación del estilo de la plantilla, revisado por DC, MM	Correcto
10	10	10	Programación del fondo de pantalla, revisado por DC, MM	Correcto
11	10	10	Revisión de transición a siguiente tarea, siendo password correcto, revisado por DC, MM	Correcto

Nota: Muestra las reuniones con las observaciones, y las soluciones de problemas suscitado en los avances diarios

Elaborado por: Diego Canchignia y Manuel Moina.

2.2.1.1.5. Sprint Review

Se registra las observaciones de los problemas suscitados en cada tarea ejecutada del Sprint terminado, el total de días utilizados, y los días de prórroga.

Tabla 11.

Sprint Review del Sprint 1

Sprint Review							
N° tarea	Id Tarea	Descripción de tarea	Responsable	N° Días	N° Días Prorroga	Estado	Observaciones
1	T01	Control de acceso	Manuel Moina	10	1	Terminado	Campos en base de datos
2	T02	Definición de roles	Manuel Moina	10	1	Terminado	Color de fondo en menú de perfiles
3	T03	Registro de profesionales	Manuel Moina	10	1	Terminado	Ícono representativo de entidades públicas

Nota: Muestra todas las tareas terminas en el sprint 1, y las observaciones suscitadas en el desarrollo
Elaborado por: Diego Canchignia y Manuel Moina.

Nota: El desarrollo y registro de todas las tareas de los Sprint 1,2 y 3, favor revisar en los Entregables de Scrum.

2.2.1.1.6. Sprint Retrospective

Se registra las observaciones de los problemas suscitados en cada tarea ejecutada y las soluciones en el Sprint terminado, el total de días utilizados, y los días de prórroga. Además es un breve resumen del Sprint que sirve para no cometer el mismo error a desarrollos futuros.

Tabla 12.

Sprint Retrospective del Sprint 1

Sprint Retrospective								
N° tarea	Id Tarea	Descripción de tarea	Responsable	N° Días	N° Días Prorroga	Estado	Observaciones	Solución
1	T01	Control de acceso	Manuel Moina	10	1	Terminado	Campos en base de datos	Se agrega campo para profesionales
2	T02	Definición de roles	Manuel Moina	10	1	Terminado	Color de fondo en menú de perfiles	Cambio de color azul acorde a la página oficial de HEE
3	T03	Registro de profesionales	Manuel Moina	10	1	Terminado	Ícono representativo de entidades públicas	Agregación de ícono acorde a la página oficial de HEE

Nota: Muestra el resumen de las tareas terminadas

Elaborado por: Diego Canchignia y Manuel Moina.

Nota: El desarrollo y registro de todas las tareas de los Sprint 1,2 y 3, favor revisar en los Entregables de Scrum.

2.3. Diseño de diagramas

Para entender los requerimientos del proyecto que se pretende desarrollar, se sostendrá con el apoyo de diagramas que son fundamentales para entender el funcionamiento de la aplicación web.

Los diagramas que se utilizarán para este proyecto están los siguientes:

- Diagrama de caso de uso
- Diagrama de secuencia
- Diagrama de clases

2.3.1. Diagrama de casos de uso

2.3.1.1. Control de acceso

Tabla 13.

Tabla del diagrama de casos de uso del control de acceso

Caso de uso:	Control de Acceso
Actor Principal:	Usuario de la aplicación
Objetivos en contexto:	Permitir acceder a la aplicación solo si el usuario tiene una cuenta activa, caso contrario restringir el uso. Permitir recuperar cambiar clave de usuario en el caso de olvido.
Precondiciones:	Es necesario que las credenciales de usuario estén registradas en la aplicación, es decir que el administrador de la aplicación debió crear la cuenta de usuario.
Disparador:	El usuario decide ingresar a la aplicación.
Escenario 1:	<ol style="list-style-type: none"> Administrador: Crea cuentas de usuario y las habilita. Usuario de la aplicación: Ingresa a la página de autenticación que despliega la aplicación Usuario de la aplicación: Registra el nombre de usuario y la clave. Usuario de la aplicación: Presiona el botón Iniciar sesión. Usuario de la aplicación: Observa la notificación de la aplicación la cual señala si las credenciales son correctas o incorrectas y da el acceso o lo restringe.
Escenario 2:	<ol style="list-style-type: none"> Usuario de la aplicación: Ingresa a la página de autenticación que despliega la aplicación Usuario de la aplicación: No recuerda la clave de usuario. Usuario de la aplicación: Da clic en el link ¿No recuerdas tu Usuario y/o Clave? Usuario de la aplicación: Ingresa la dirección correo electrónico asociado a la cuenta de usuario en el formulario que desea registrar nueva clave. Usuario de la aplicación: Recibe un correo de la aplicación de partes operatorios el cual le da un link para ingresar una nueva clave. Usuario de la aplicación: Ingresa nueva clave, y confirma el cambio. Usuario de la aplicación: Ingresa al sistema con las nuevas credenciales.
Excepciones:	<ol style="list-style-type: none"> Clave incorrecta: La aplicación despliega un mensaje indicando que el usuario y/o contraseña son incorrectos. Usuario no está registrado en la aplicación: La aplicación despliega un mensaje indicando que el usuario y/o contraseña son incorrectos. Usuario olvidó su clave y/o nombre de usuario: El usuario de la aplicación debe ponerse en contacto con el administrador.
Prioridad:	Alta, es necesario asegurar el acceso a la aplicación.
Cuando estará disponible:	En el primer incremento.
Frecuencia de uso:	Varias veces al día.
Canal para el actor:	A través de la interfaz de autenticación del sistema
Actores secundarios:	Administrador
Canales para los actores secundarios:	Módulo de administración de usuarios
Aspectos pendientes:	Agregar módulos de administración de usuarios

Nota: Muestra los requerimientos que se solventarán, y los actores que estarán interactuando en la aplicación.

Elaborado por: Diego Canchignia y Manuel Moina.

El siguiente diagrama de casos de uso representa graficamente la tabla de control de acceso, pauta a los actores y los procesos que interviene en las seguridades para el ingreso la aplicación web.

Diagrama de casos de uso del control de acceso

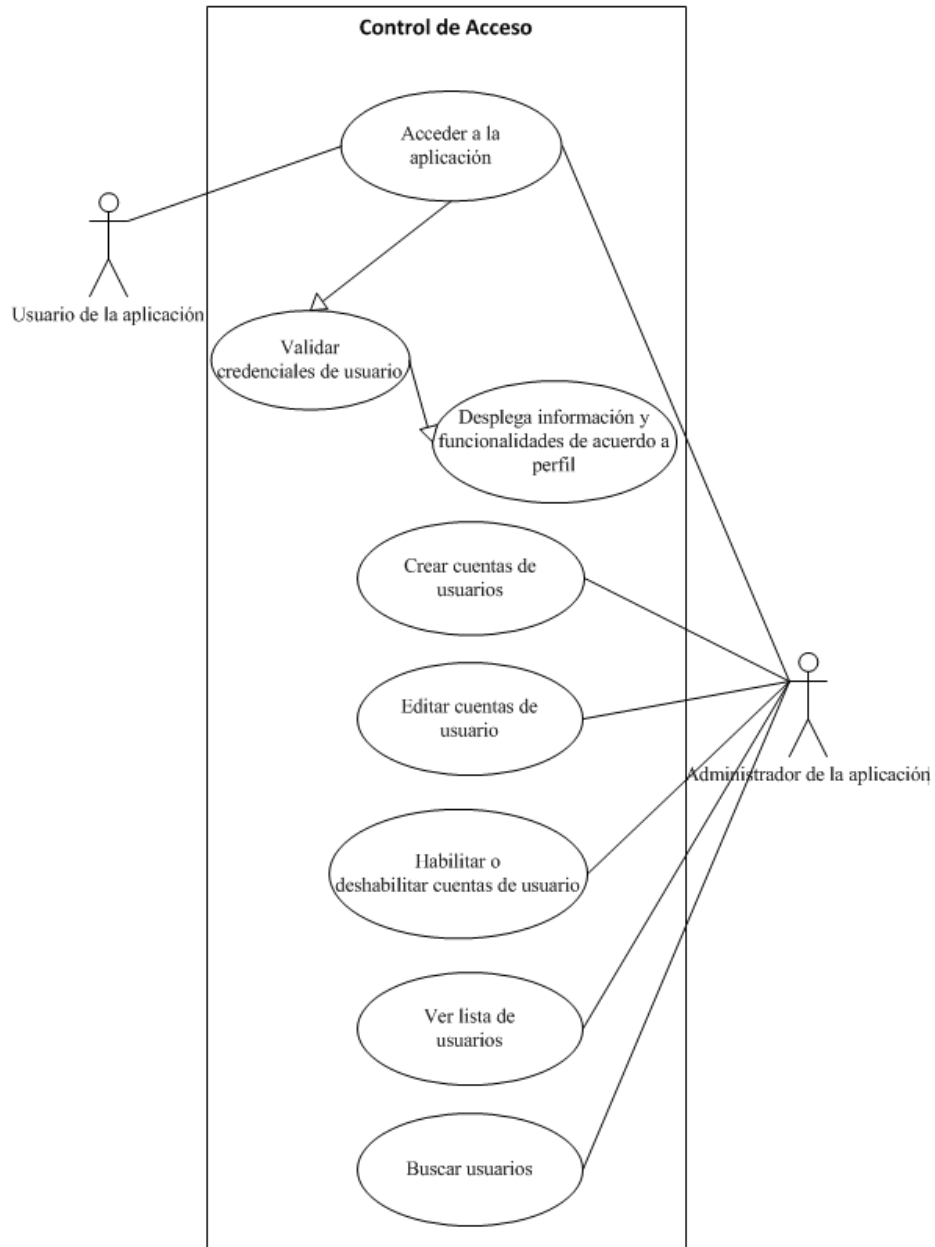


Figura 11. Muestra los requerimientos que se solventarán, y los actores que estarán interactuando en la aplicación

Elaborado por: Diego Canchignia y Manuel Moina.

2.3.1.2. Definición de roles

Tabla 14.

Tabla del diagrama de casos de uso de la definiciones de roles

Caso de uso:	Definición de roles
Actor Principal:	Administrador de la aplicación
Objetivos en contexto:	Permitir acceder a los usuarios a la información o funcionalidades que provee la aplicación de acuerdo al rol de usuario registrado en la cuenta.
Precondiciones:	Es necesario que la cuenta de usuario esté relacionada a un rol de acuerdo a las funciones que realiza el usuario en la aplicación.
Disparador:	El usuario decide ingresar a la aplicación.
Escenario:	<ol style="list-style-type: none">1. Administrador: Crea, edita o deshabilita roles de usuario.2. Usuario de la aplicación: Ingresa a la página de autenticación que despliega la aplicación3. Usuario de la aplicación: Registra el nombre de usuario y la clave.4. Usuario de la aplicación: Presiona el botón “Iniciar Sesión” luego la aplicación despliega solo la información y funciones a las que tiene acceso el rol de usuario de la cuenta.
Excepciones:	<ol style="list-style-type: none">1. Usuario de la aplicación no tiene cuenta de usuario.2. Rol de usuario esta deshabilitado.
Prioridad:	Alta, es necesario personalizar las funciones que puede hacer el usuario de acuerdo a su perfil.
Cuando estará disponible:	En el primer incremento.
Frecuencia de uso:	Varias veces al día.
Canal para el actor:	A través de la interfaz de autenticación del sistema
Actores secundarios:	Usuarios de la aplicación
Canales para los actores secundario	Módulo de administración de usuarios
Aspectos pendientes:	N/A

Nota: Muestra los requerimientos que se solventaran, y los actores que estarán interactuando en la aplicación

Elaborado por: Diego Canchignia y Manuel Moina.

El siguiente diagrama de casos de uso representa graficamente la tabla de definición de roles, pauta a los actores y los procesos que interviene para la asignación de roles para aplicación web.

La manipulación de la aplicación web mediante perfiles es importante, puesto que cada perfil podrá ingresar a un menú asignado, y según el menú registrará la información en el proceso del parte operatorio.

Diagrama de casos de uso de la definiciones de roles

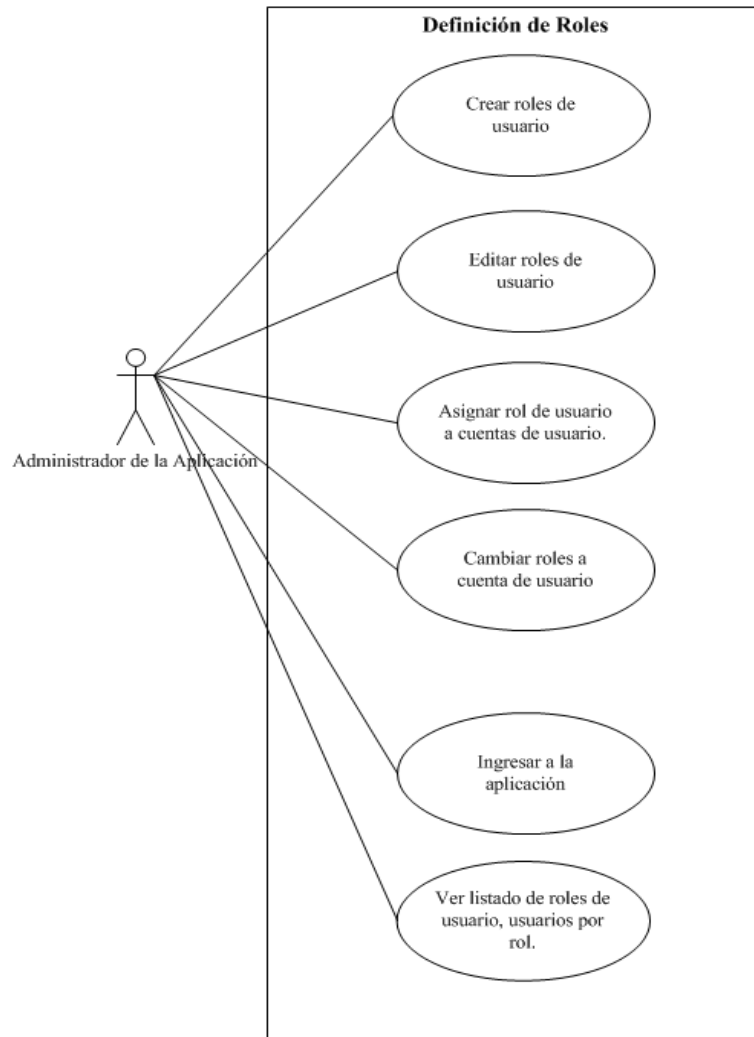
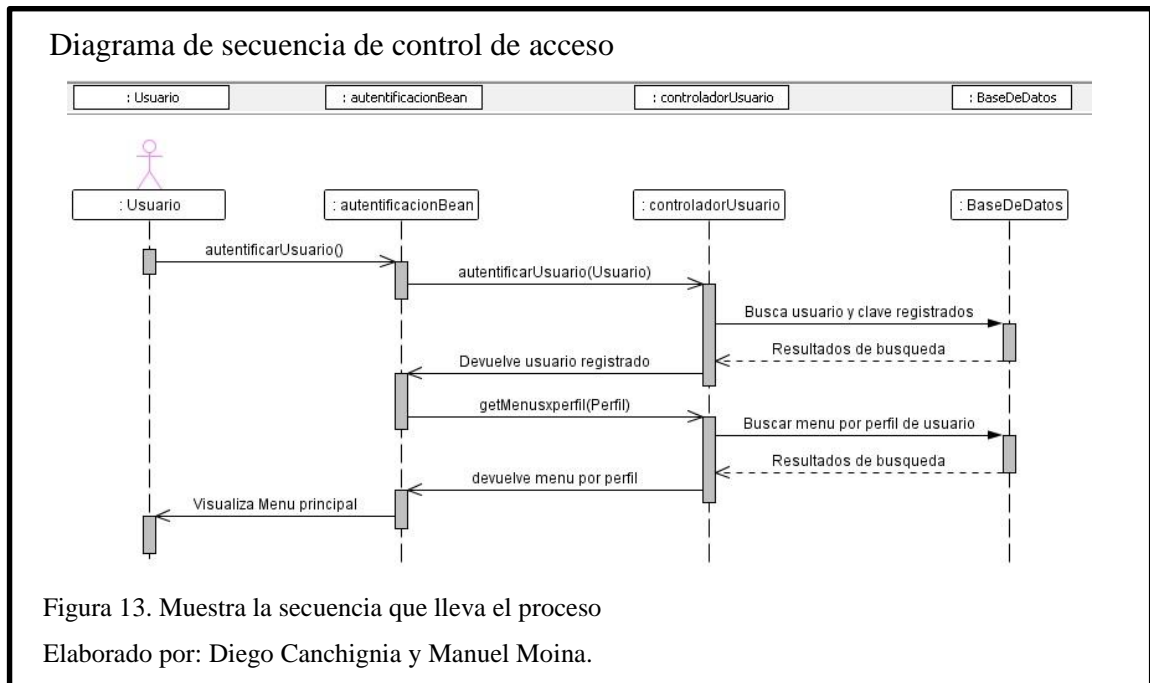


Figura 12. Muestra los requerimientos que se solventaran, y los actores que estarán interactuando en la aplicación

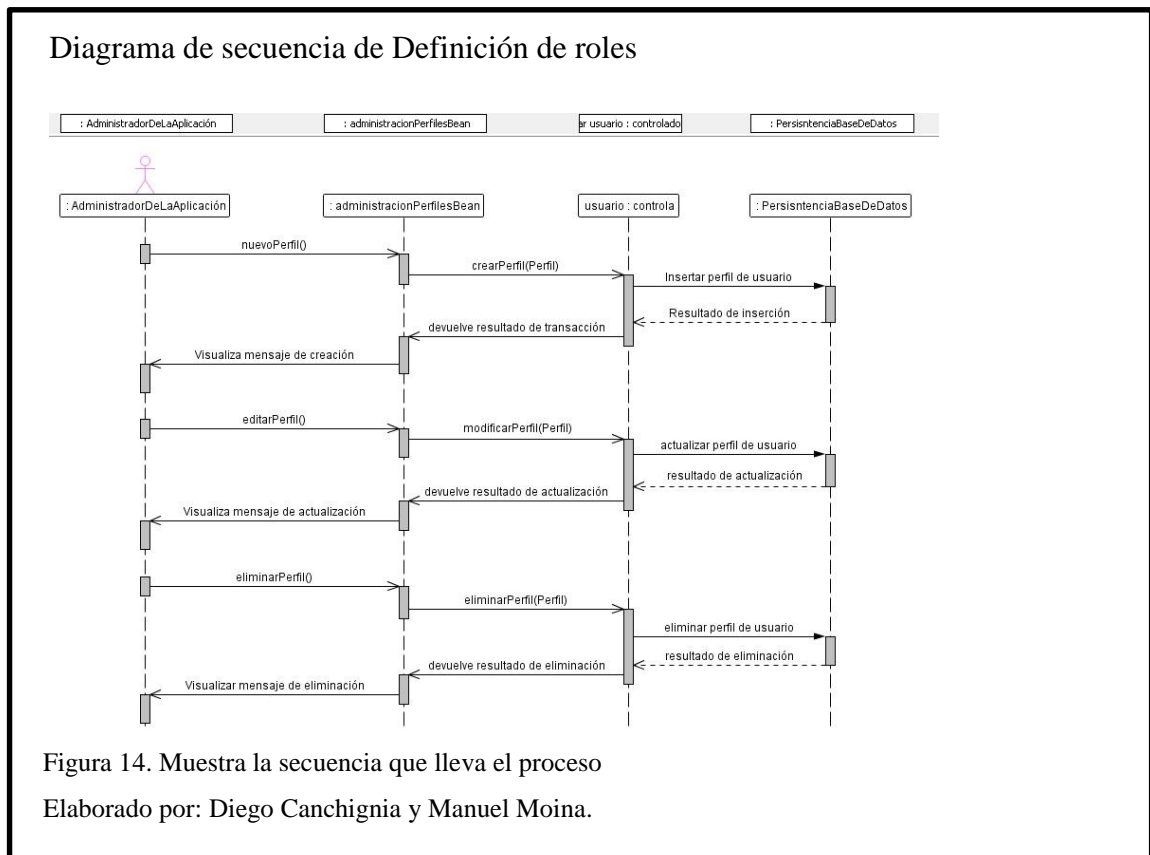
Elaborado por: Diego Canchignia y Manuel Moina.

2.3.2. Diagramas de secuencia

2.3.2.1. Control de acceso



2.3.2.2. Definiciones de Roles



2.3.3. Diagramas de Clase

Diagrama de clases

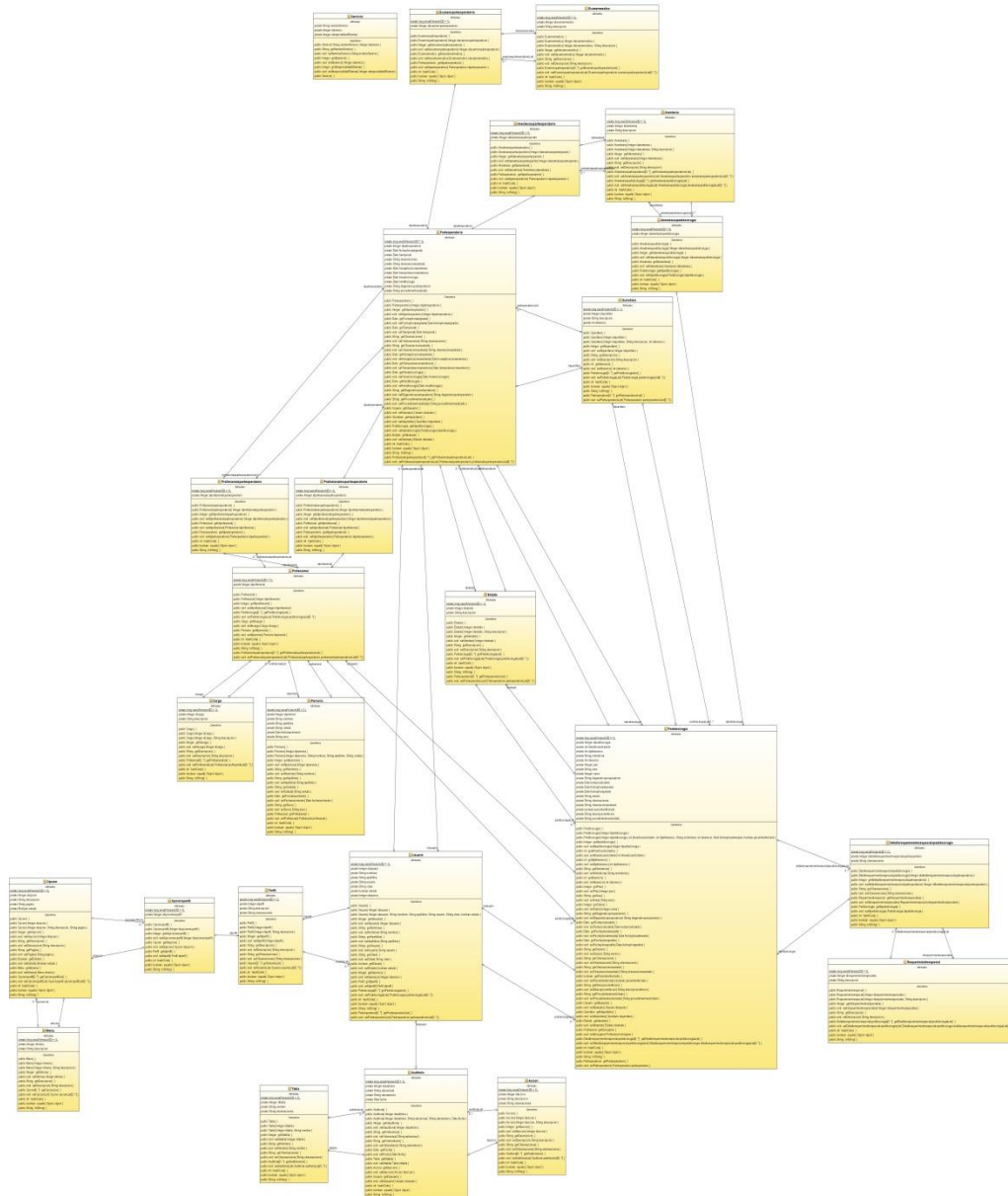


Figura 15. Muestra todas las clases que tendrá la aplicación web

Elaborado por: Diego Canchignia y Manuel Moina.

2.3.3.1. Diagrama de clases (Controladores)

Diagrama de clases de controladores 1



Figura 16. Muestra las clases: Opción, Abstract, Profesional por parte operatorio, Parte Operatorio

Elaborado por: Diego Canchignia y Manuel Moína.

Diagrama de clases de controladores 2

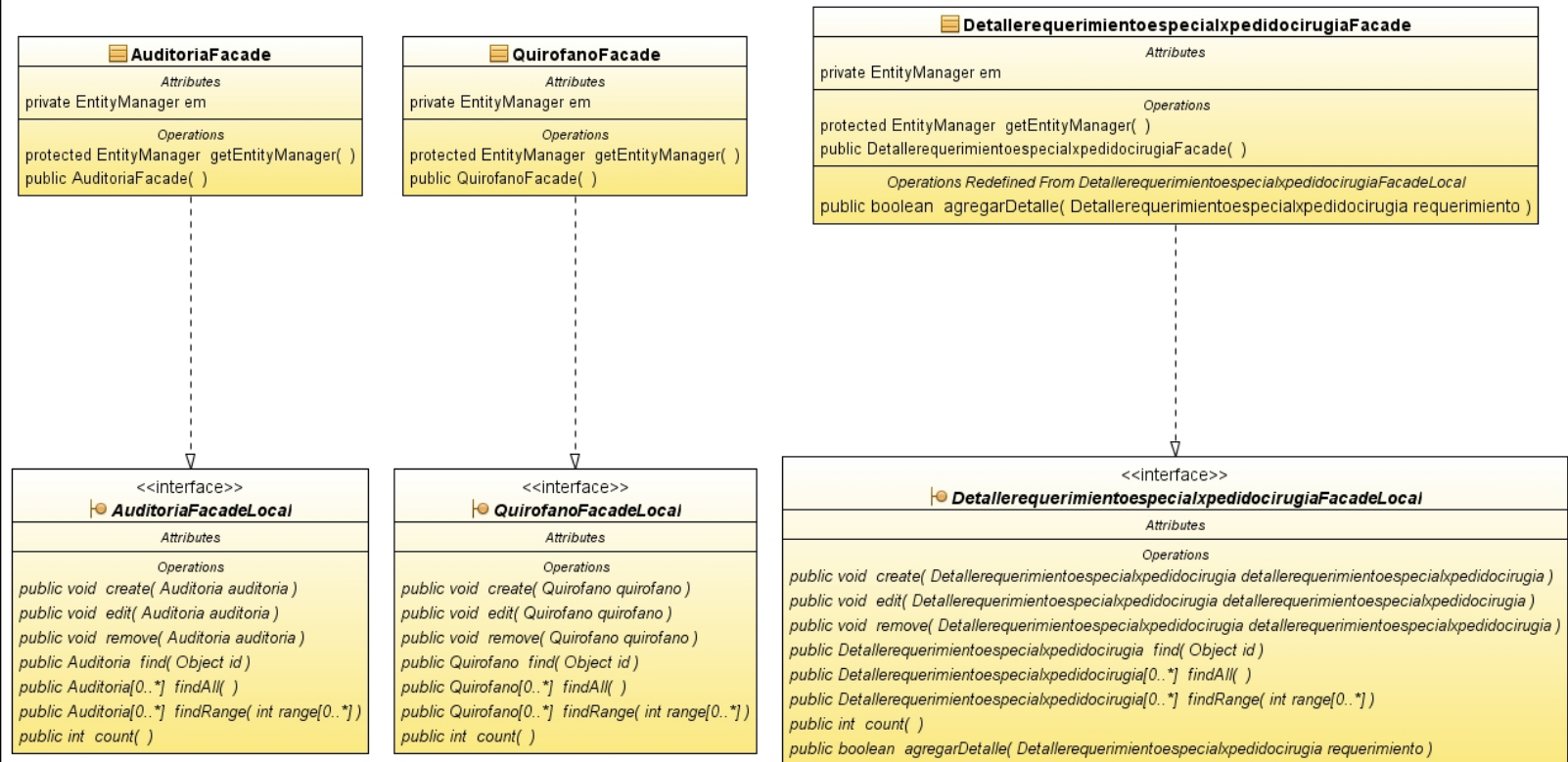


Figura 17. Muestra las clases: Auditoría, Quirófano, Detalle requerimiento especial por pedido de cirugía

Elaborado por: Diego Canchignia y Manuel Moina.

2.4. Diseño de la Base de Datos

2.4.1. Diseño Lógico

Diagrama lógico de la base de datos

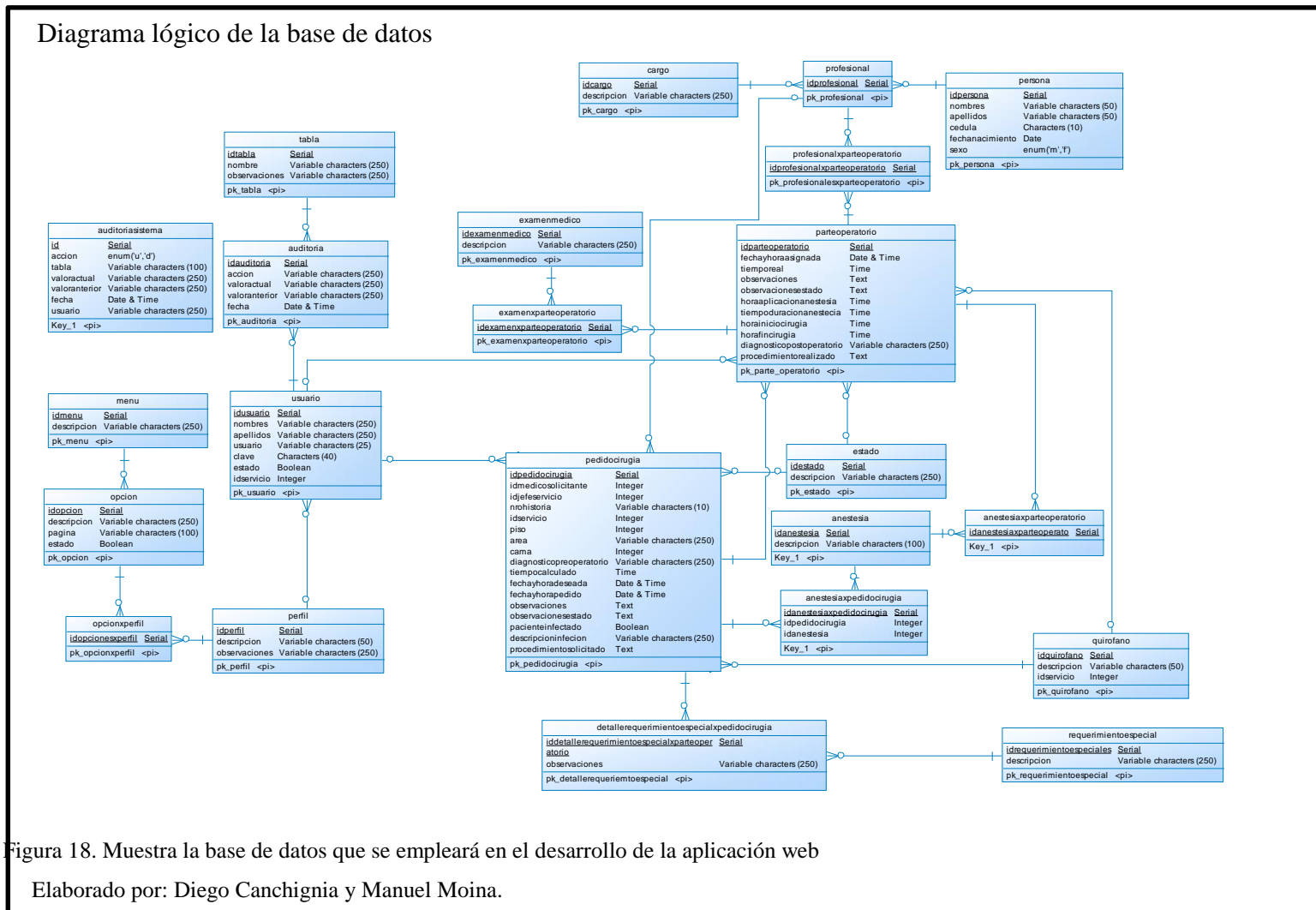


Figura 18. Muestra la base de datos que se empleará en el desarrollo de la aplicación web

Elaborado por: Diego Canchignia y Manuel Moina.

2.4.2. Diseño Físico

Diagrama físico de la base de datos

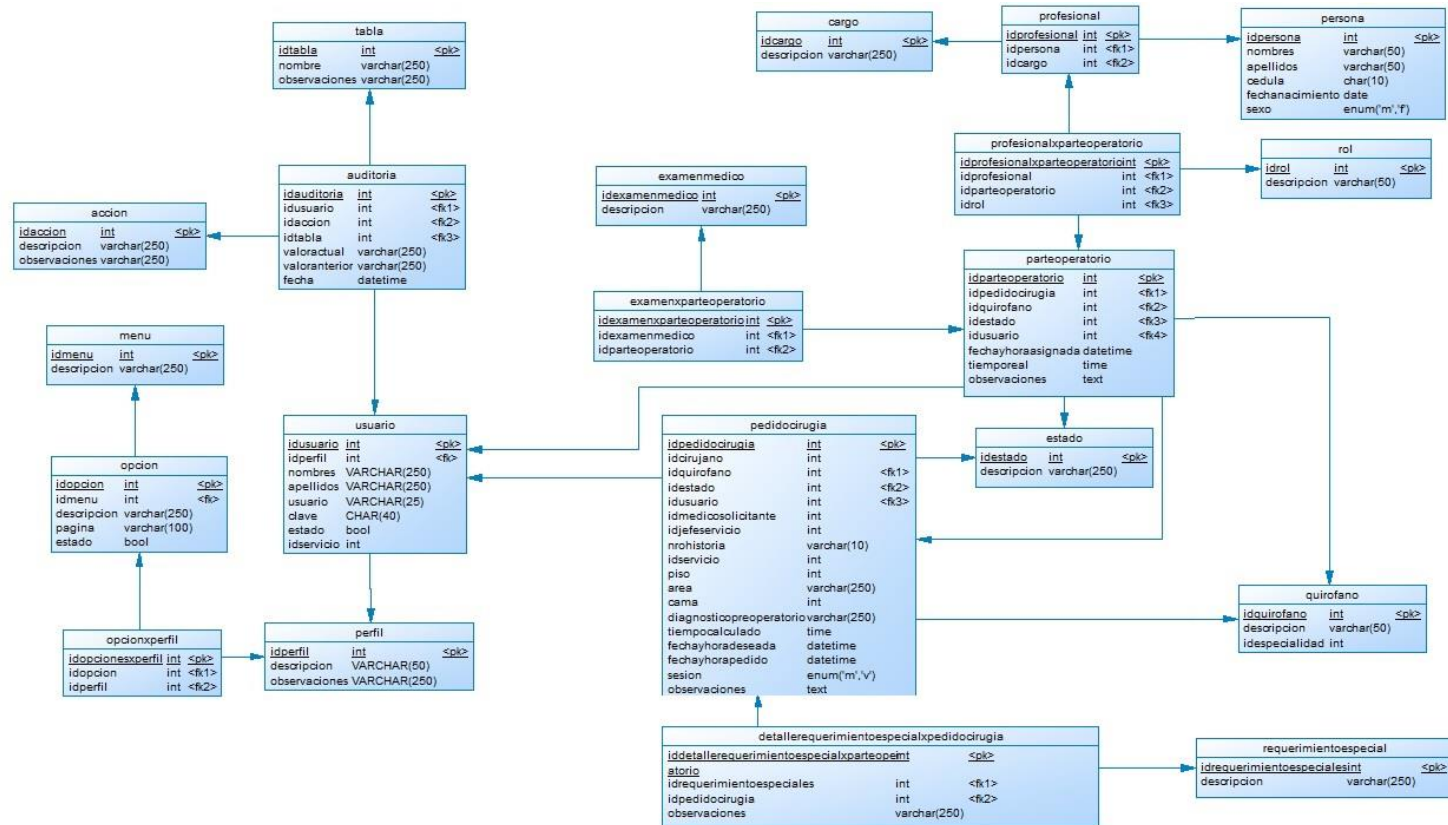


Figura 19. Muestra la base de datos que se empleará en el desarrollo de la aplicación web

Elaborado por: Diego Canchignia y Manuel Moína.

2.4.3.1. Usuario

Tabla 15.

Tabla del diccionario de datos de usuario

Usuario					
Contiene información de los usuarios de la aplicación					
Nombre del campo	Tipo de dato	Obligatorio S/N	Restricción/Constraint	Valor por defecto	Descripción/Comentario
idusuario	int	S	Primary Key		Identificador único de usuario
idperfil	int	S	Foreign key		Clave forenea para relacionar un usuario con un perfil
nombres	varchar(50)	S	Unique key		Nombre del usuario registrado en la aplicación
apellidos	varchar(50)	S			Apellidos del usuario registrado en la aplicación
usuario	varchar(50)	S	Unique key		Nombre o alias de usuario registrado en la aplicación
clave	varchar(50)	S	N/A		Clave de usuario de la aplicación
estado	boolean	S	N/A	1	Establece el estado del usuario(Activo, bloqueado)
idespecialidad	int	N	N/A		Relaciona al usuario de la aplicación a una especialidad registrada en el sistema shaman
Constraints					
Nombre del constraint	Tipo de constraint	Definición	Descripción/Comentario		
pk_usuario	Primary key	primary key (IDUSUARIO)			
fk_usuario_perfil	Foreign key	alter table usuario add constraint fk_usuario_perfil foreign key (idperfil) references perfil (idperfil) on delete restrict on update restrict;			
uk_usuario	Unique key	create unique index uk_usuario on usuario(usuario);			
uk_nombres_apellidos	Unique key	create unique index uk_nombres_apellidos on usuario(nombres, apellidos);			

Nota: Muestra los atributos de los datos, clave primaria y/o secundaria, y relación con otras tablas

Elaborado por: Diego Canchignia y Manuel Moina.

2.4.3.2. Perfil

Tabla 16.

Tabla del diccionario de datos de perfil

Perfil					
Contiene información de los perfiles que pueden tener los usuarios de la aplicación					
Nombre del campo	Tipo de dato	Obligatorio S/N	Restricción/Constraint	Valor por defecto	Descripción/Comentario
idperfil	int	S	Primary Key		Identificador único de perfil
descripcion	varchar(50)	S	Unique key		Descripción del perfil de usuario
observaciones	varchar(250)	N	N/A		Observaciones de perfil de usuario
Constraints					
Nombre del constraint	Tipo de constraint	Definición	Descripción/Comentario		
pk_perfil	Primary key	primary key (idperfil)			
uk_descripcion_perfil	Unique key	uk_descripcion_perfil on perfil(descripcion);			

Nota: Muestra los atributos de los datos, clave primaria y/o secundaria, y relación con otras tablas

Elaborado por: Diego Canchignia y Manuel Moina.

2.4.3.3. Opción por perfil

Tabla 17.

Tabla del diccionario de datos de opción de perfil,

Opcionxperfil	Contiene información de los permisos que puede tener un perfil de usuario				
Nombre del campo	Tipo de dato	Obligatorio S/N	Restricción/Constraint	Valor por defecto	Descripción/Comentario
idopcionxperfil	int	S	Primary Key		Identificador único de opción por perfil
idopcion	int	S	Foreign key		Identificador de opción
idperfil	int	S	Foreign key		Identificador de perfil
Constraints					
Nombre del constraint	Tipo de constraint	Definición	Descripción/Comentario		
pk_opcionxperfil	Primary key	primary key (idopcionxperfil)			
fk_opcionxperfil_perfil	Foreign key	alter table permiso add constraint fk_opcionxperfil_perfil foreign key (idperfil) references perfil			
fk_opcionxperfil_opcion	Foreign key	alter table opcion add constraint fk_opcionxperfil_opcion foreign key (idopcion) references			

Nota: Muestra los atributos de los datos, clave primaria y/o secundaria, y relación con otras tablas
Elaborado por: Diego Canchignia y Manuel Moina.

2.4.3.4. Opción

Tabla 18.

Tabla del diccionario de datos de opción

Opcion	Contiene información de las opciones que pueden tener los usuarios de la aplicación				
Nombre del campo	Tipo de dato	Obligatorio S/N	Restricción/Constraint	Valor por defecto	Descripción/Comentario
idopcion	int	S	Primary Key		Identificador único de permiso
idmenu	int	S	Foreign key		Relaciona al la opción a un menu
descripcion	varchar(50)	S	Unique key		Descripción del permiso
pagina	varchar(100)	S	N/A		Nombre de la página a la cual se tiene acceso mediante esta opción
estado	boolean	N	N/A	true	Observaciones de permiso
Constraints					
Nombre del constraint	Tipo de constraint	Definición	Descripción/Comentario		
pk_opcion	Primary key	primary key (idopcion)			
fk_opcion_menu	Foreign key	alter table opcion add constraint fk_opcion_menu foreign key (idmenu) references menu			
uk_pagina	Unique key	create unique index uk_pagina on opcion(pagina);			
uk_descripcion_opcion	Unique key	create unique index uk_descripcion_opcion on opcion(descripcion);			

Nota: Muestra los atributos de los datos, clave primaria y/o secundaria, y relación con otras tablas
Elaborado por: Diego Canchignia y Manuel Moina.

2.5. Diseño de la interfaz de la aplicación

2.5.1. Pantalla de logeo para el control de acceso

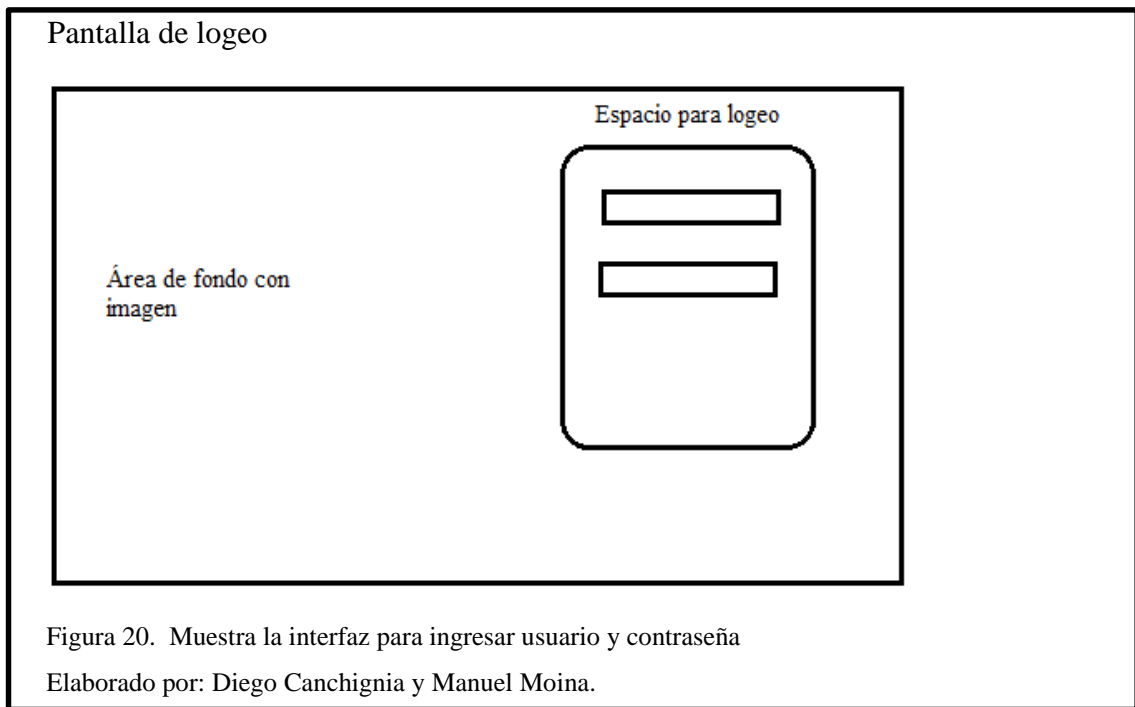


Figura 20. Muestra la interfaz para ingresar usuario y contraseña

Elaborado por: Diego Canchignia y Manuel Moina.

2.5.2. Menú por perfiles

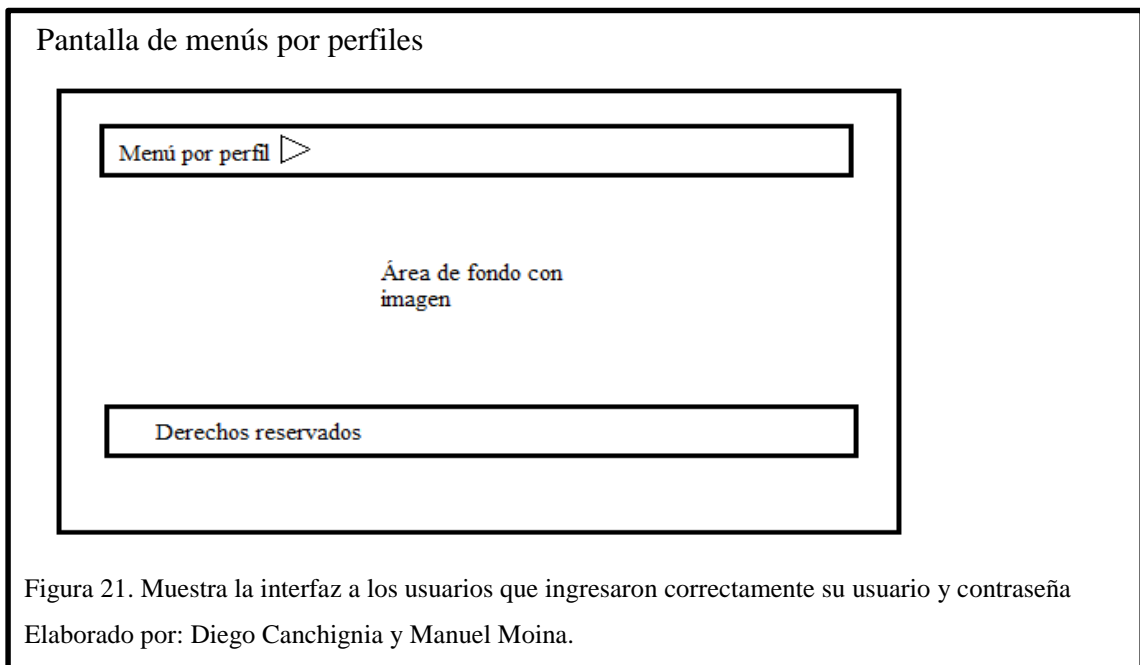
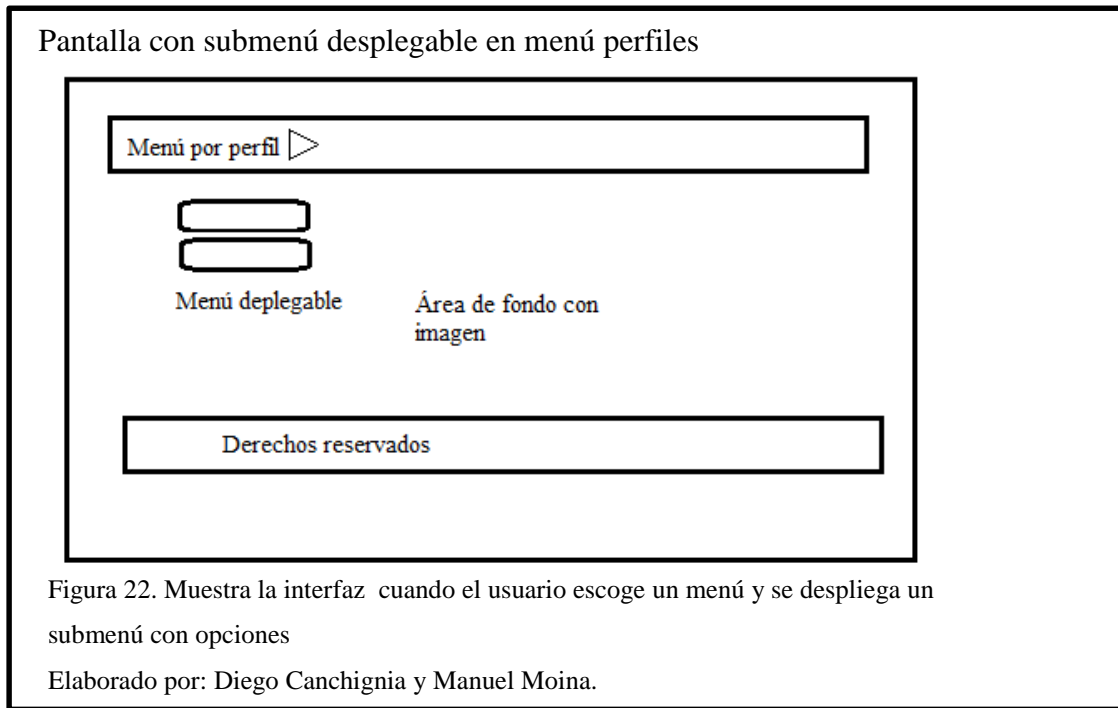


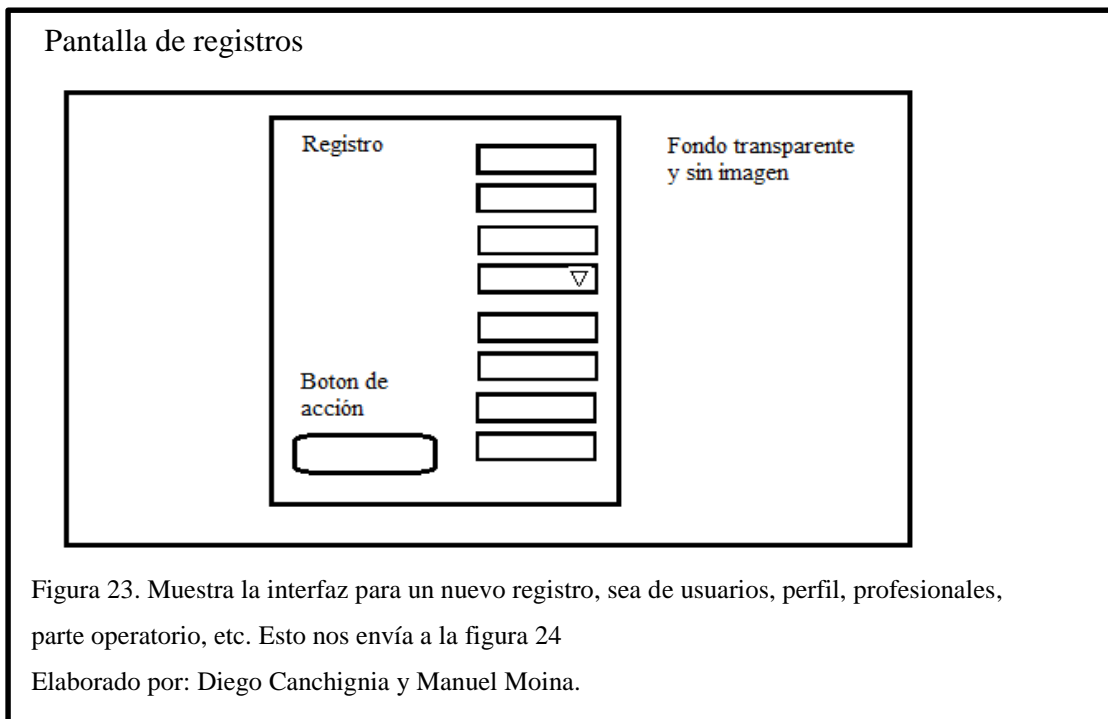
Figura 21. Muestra la interfaz a los usuarios que ingresaron correctamente su usuario y contraseña

Elaborado por: Diego Canchignia y Manuel Moina.

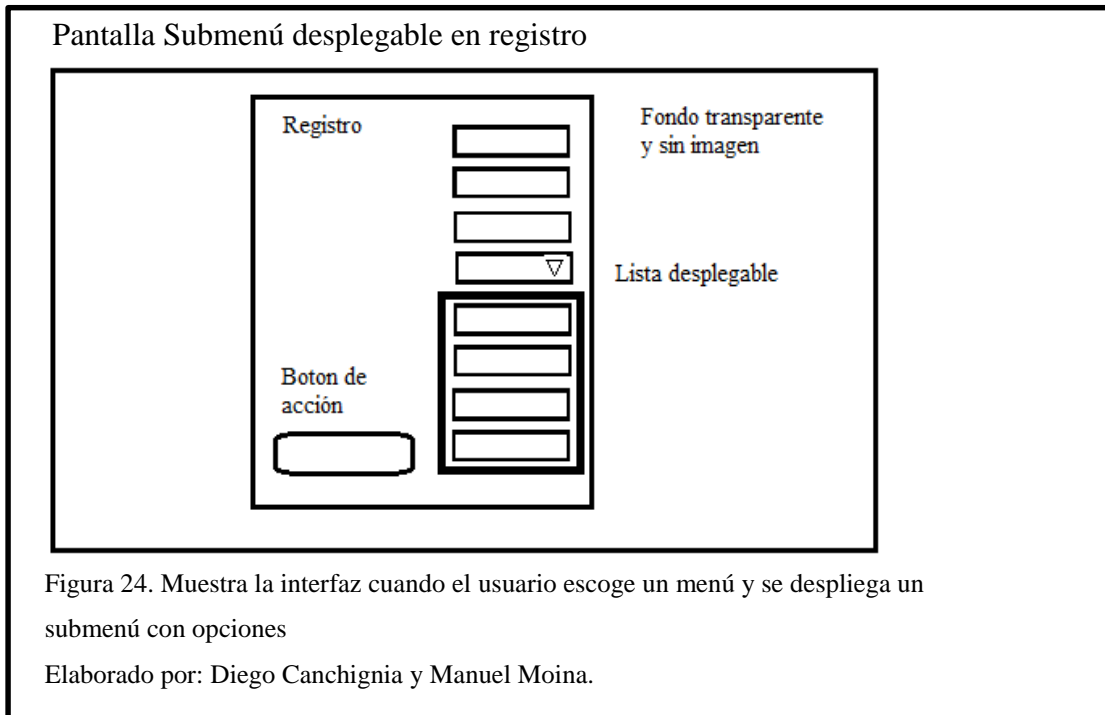
2.5.3. Submenú desplegable en menú de perfiles



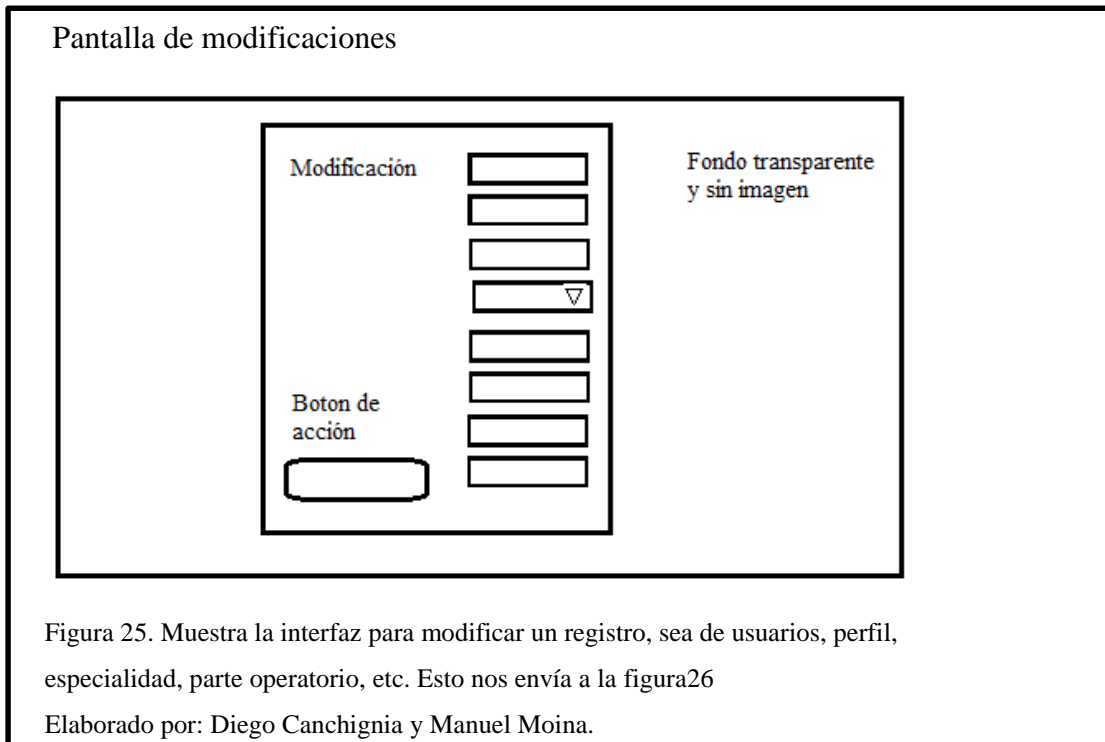
2.5.4. Registros usuarios, perfiles, profesionales



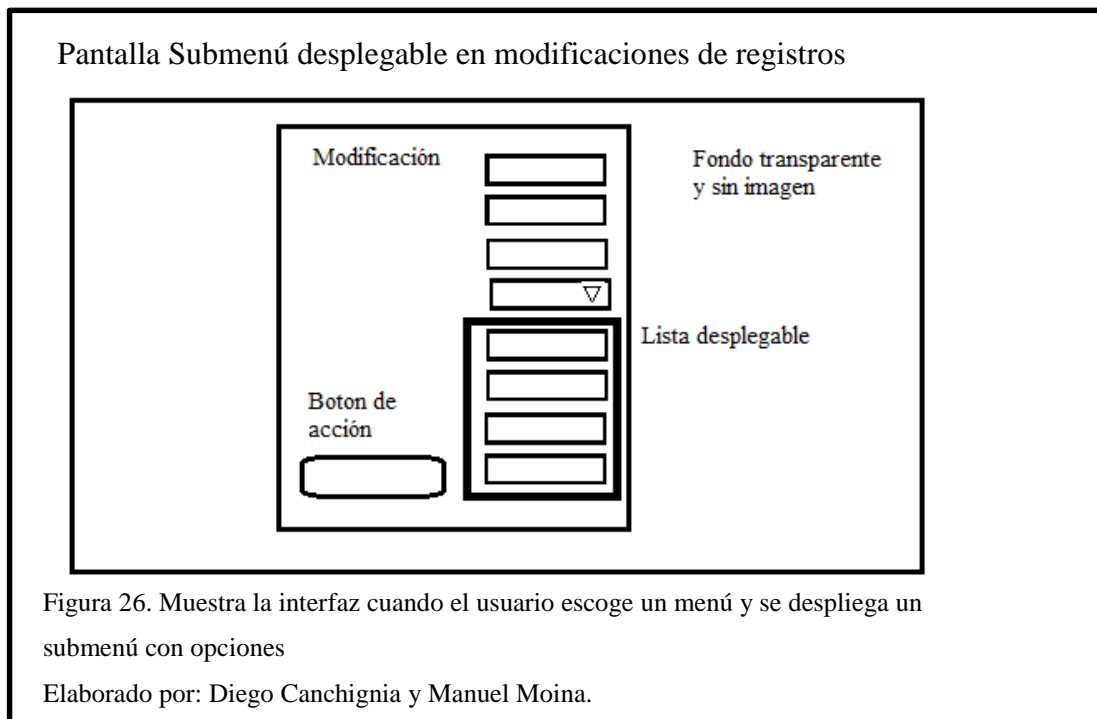
2.5.5.- Submenú desplegable en registros de usuarios, perfiles, profesionales



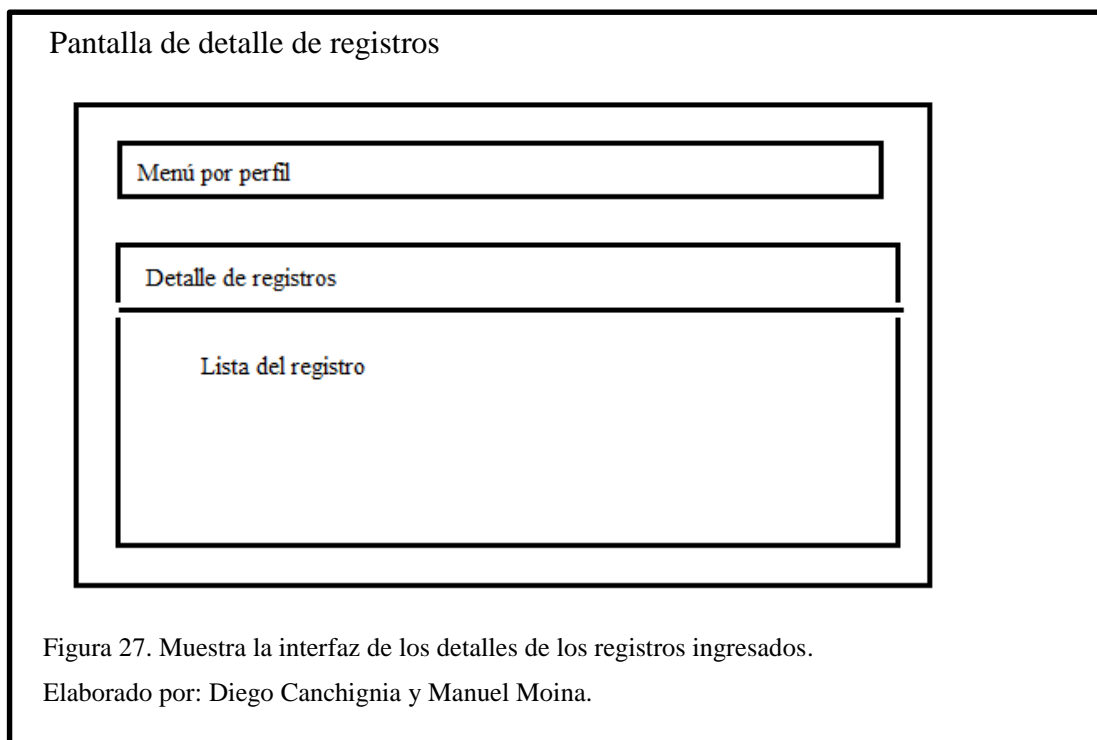
2.5.6. Modificaciones de registros de usuarios, perfiles, profesionales



2.5.7. Submenú desplegable en modificaciones de registros



2.5.8.- Detalle de registros



CAPÍTULO 3

GENERACIÓN DEL SISTEMA INFORMÁTICO

3.1. Método Esquemático

Diagrama esquemático

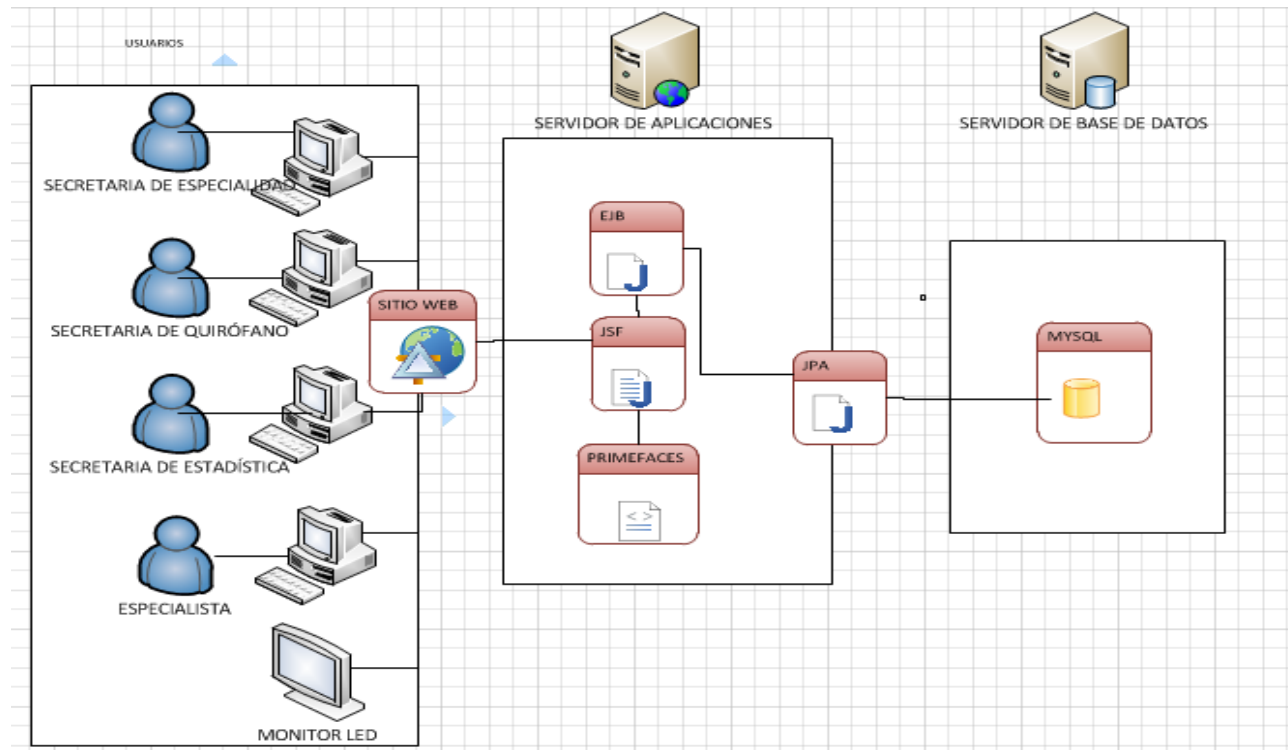


Figura 28. Muestra la arquitectura que llevara a nivel global la aplicación en el Hospital de Especialidades Eugenio Espejo
Elaborado por: Diego Canchignia y Manuel Moina.

3.2. Diagrama de Proceso del Parte Operatorio

El proceso general de la aplicación web para administrar y gestionar los partes operatorios.

Subproceso 1: La secretaría de especialidad registra la solicitud del pedido de cirugía, con información del paciente, activa subproceso dos.

Subproceso 2: La junta médica evalúa los pedidos de cirugías y decide si aprueba o rechaza la solicitud.

Sí, se aprueba la cirugía, activa subproceso 3.

No, se rechaza solicitud, se registra la observación del rechazo y se notifica, la notificación se observa reflejada en secretaría de especialidad.

Subproceso 3: La secretaría de quirófanos evalúa nuevamente el pedido de cirugía que fue previamente analizado y aprobado en el subproceso 2, además verifica si hay disponibilidad de quirófanos, profesionales de salud, insumos médicos, etc. Y decide lo siguiente:

Sí, se aprueba la cirugía, y se agenda el quirófano, los profesionales de la salud, insumos médicos, activa subproceso 4.

No, se rechaza solicitud, se registra la observación del rechazo y se notifica, la notificación se observa reflejada en secretaría de especialidad.

Subproceso 4: Una vez agendado se notifica a los servicios solicitantes la aprobación y se publica las cirugías, activa subproceso 5.

Subproceso 5: En el quirófano central el paciente es sometido a la intervención quirúrgicas, se activa el subproceso 6.

Subproceso 6: En la secretaría de quirófano se registra el parte operatorio, las observaciones de la cirugía y la hoja de anestesia.

Subproceso 7: Fin del proceso.

Este proceso se encuentra ilustrado en la figura 7 que se encuentra a continuación.

Diagrama de procesos del parte operatorio

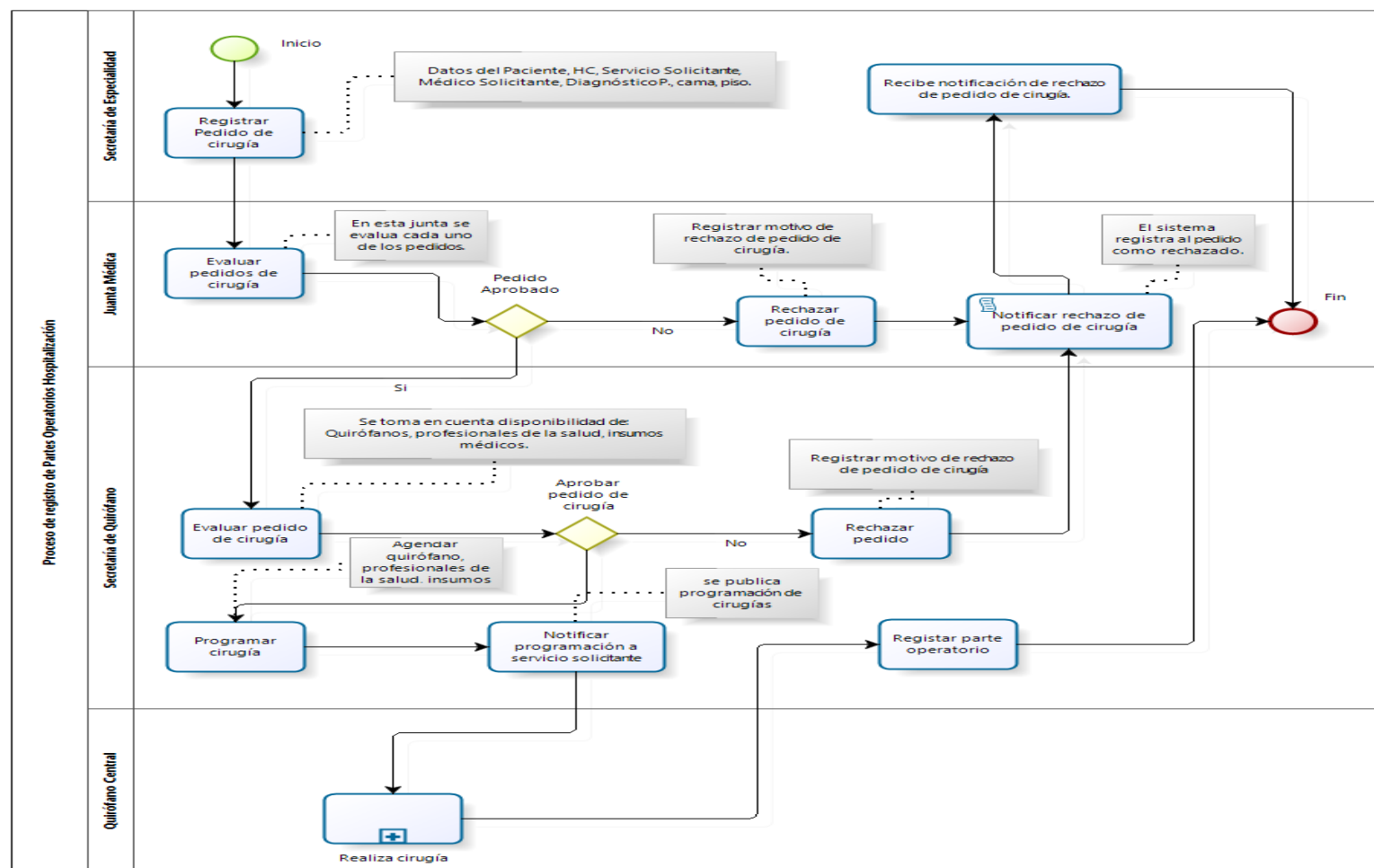


Figura 29. Muestra el proceso que integra el parte operatorio desde la solicitud, hasta el registro, aprobado por la Líder del Departamento de Estadística (Subrogante), ver anexo 5

Elaborado por: Diego Canchignia y Manuel Moina.

3.3. Desarrollo del código de la aplicación

3.3.1. Control de acceso

Con el fin de garantizar la seguridad de acceso a la aplicación web, las claves de los usuarios están encriptados en MD5¹⁹ y nuevamente en SHA1²⁰, podemos observar los métodos de encriptación:



Los métodos de encriptación usan la librería: `java.security.MessageDigest`;
`package metodos`;

```
import java.math.BigInteger;
```

```
import java.security.MessageDigest;
```

```
public class metodos {
```

```
    public metodos() { }
```

```
    public String generarMd5(String entrada){
```

```
        try { MessageDigest md = MessageDigest.getInstance("MD5");
```

19. MD5 (Message-Digest Algorithm 5) Algoritmo de Resumen del Mensaje 5, es un algoritmo de reducción criptoFigura de 128 bits.

20. SHA1 (Secure Hash Algorithm,) Algoritmo de Hash Seguro, algoritmos que se utilizan principalmente para verificar la integridad de archivos y comunicaciones digitales y reducción criptoFigura de 160 bits

```

        byte[] resultado = md.digest(entrada.getBytes());
        BigInteger numero = new BigInteger(1, resultado);
        String textoHash = numero.toString(16);
        // Now we need to zero pad it if you actually want the full 32 chars.
        while (textoHash.length() < 32) {
            textoHash = "0" + textoHash;
        }

        return textoHash;
    } catch (Exception e) {
        System.out.println("Error al encriptar cadena: "+e.getMessage());
        return "";
    }
}

public String generarSha1(String entrada){
    try {
        MessageDigest mDigest = MessageDigest.getInstance("SHA1");
        byte[] resultado = mDigest.digest(entrada.getBytes());
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < resultado.length; i++) {
            sb.append(Integer.toString((resultado[i] & 0xff) + 0x100, 16).substring(1));
        }
        return sb.toString();
    } catch (Exception e) {
        System.out.println("Error al generar Sha1: "+e.getMessage());
        return "";
    }
}
}

```

Una vez creado el usuario de la aplicación, el mismo se puede autenticar en la aplicación para usar todas las funcionalidades a las que le permite su perfil, para ello

debe registrar su nombre de usuario y contraseña como se ve en la página e autenticación de la aplicación.

```
public String autenticarUsuario(){
    metodos m = new metodos();
    String claveMD5 = m.generarMd5(usuarioActual.getClave());
    String claveSha1 = m.generarSha1(claveMD5);
    usuarioActual.setClave(claveSha1);
    usuarioActual = usuarioFacade.autenticarUsuario(usuarioActual);
    if(usuarioActual == null){
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Error", "Usuario y o
clave incorrectas")); return "";
    } else{
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_INFO, "Bienvenido ",
                usuarioActual.getIdperfil().getDescripcion()
                +": "+usuarioActual.getNombres() + " +usuarioActual.getApellidos()));
        HttpSession session = (HttpSession)
FacesContext.getCurrentInstance().getExternalContext().getSession(false);
        session.setAttribute("usuario", usuarioActual);
        generarMenu();
        return "principal.jsf?faces-redirect=true";
    } }
}
```

Además se cuenta con un filtro que valida que el usuario esté logeado, si un usuario conoce un link de la aplicación no podrá acceder sin antes estar registrado y logeado, está en una clase java.

```
public void doFilter(ServletRequest request, ServletResponse response,
    FilterChain chain) throws IOException, ServletException {
```

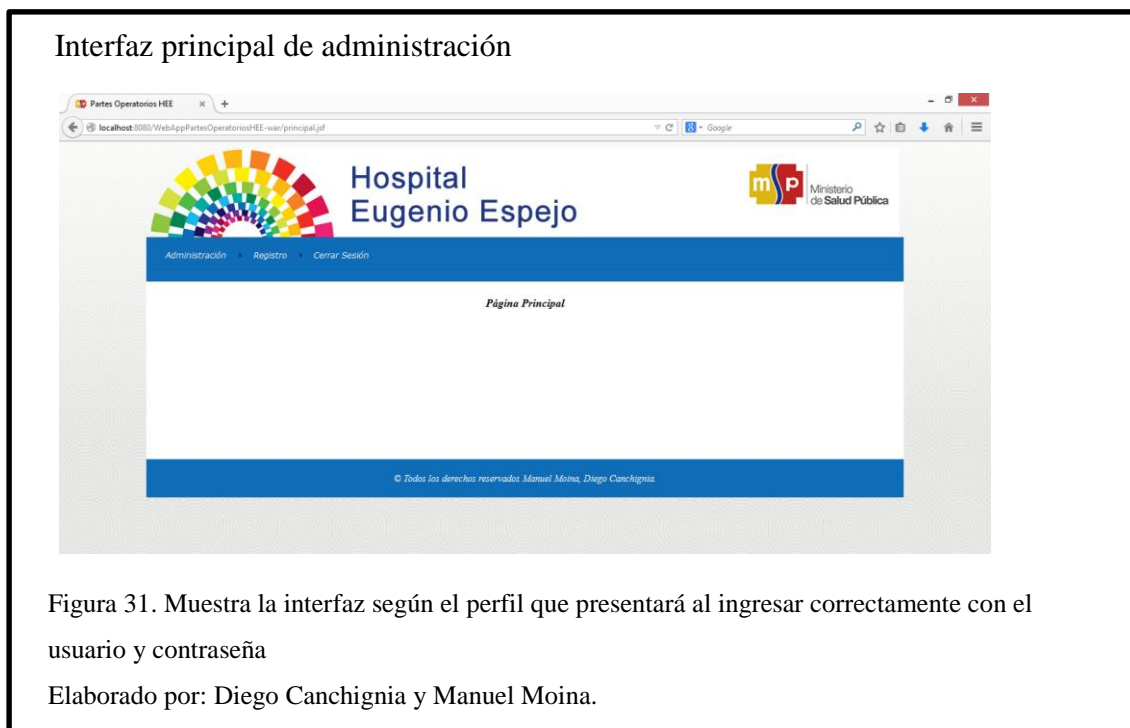
```

HttpServletRequest servletrequest = (HttpServletRequest)request;
HttpSession session =servletrequest.getSession(false);
String pagina = servletrequest.getRequestURI();
//System.out.println("pagina solicitada: "+pagina);
if(pagina.indexOf("javax.faces.resource") != -1 || pagina.indexOf("resources") !=
-1){ //System.out.println("Página sin restricción: "+pagina);
chain.doFilter(request, response);
}else if(session == null || session.getAttribute("usuario") == null){
//System.out.println("Sesión invalida. Redireccionando");
servletrequest.getRequestDispatcher("/index.jsf").forward(request, response);
return;      } //System.out.println("Acceso Autorizado.");
ry { chain.doFilter(request, response);
} catch (Exception e) {
} return; }

```

3.3.2. Definición de roles

3.3.2.1. Página principal de administración



3.3.2.2. Administración de usuarios

Interfaz de administración de usuarios



Figura 32. Muestra la interfaz para registrar usuarios

Elaborado por: Diego Canchignia y Manuel Moina.

3.3.2.3. Registro de usuarios y definición de roles

Interfaz de registro de usuario

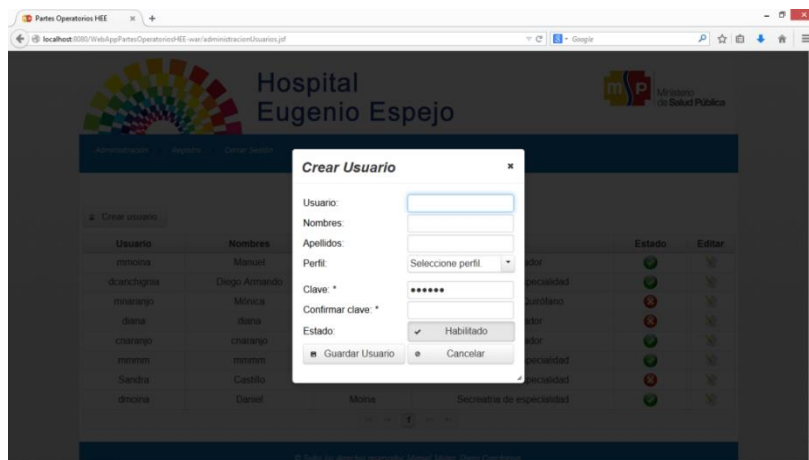


Figura 33. Muestra la interfaz para creación de nuevos usuarios y colocar su rol respectivo

Elaborado por: Diego Canchignia y Manuel Moina.

Cuando se crea un nuevo usuario, se aplica los métodos de encriptación en la clave ingresada para el nuevo usuario de la aplicación como lo podemos ver en el método.

btnNuevoUsuario().

```
public void btnNuevoUsuario(){
    metodos m = new metodos();
    String claveMD5 = m.generarMd5(nuevoUsuario.getClave());
    String claveSha1 = m.generarSha1(claveMD5);
    nuevoUsuario.setClave(claveSha1);
    boolean flag = usuarioFacade.agregarUsuario(nuevoUsuario);
    if(flag){ FacesContext.getCurrentInstance().addMessage(null,
        new FacesMessage(FacesMessage.SEVERITY_INFO, "Usuario", "Creado
Correctamente.)); inicializarUsuario();
        listaUsuarios = usuarioFacade.findAll();
        RequestContext.getCurrentInstance().execute("dlgCrearUsuario.hide()");
    } else{ FacesContext.getCurrentInstance().addMessage(null,
        new FacesMessage(FacesMessage.SEVERITY_ERROR, "Usuario", "No se
creo.));    }}
```

3.3.3. Registro de profesionales

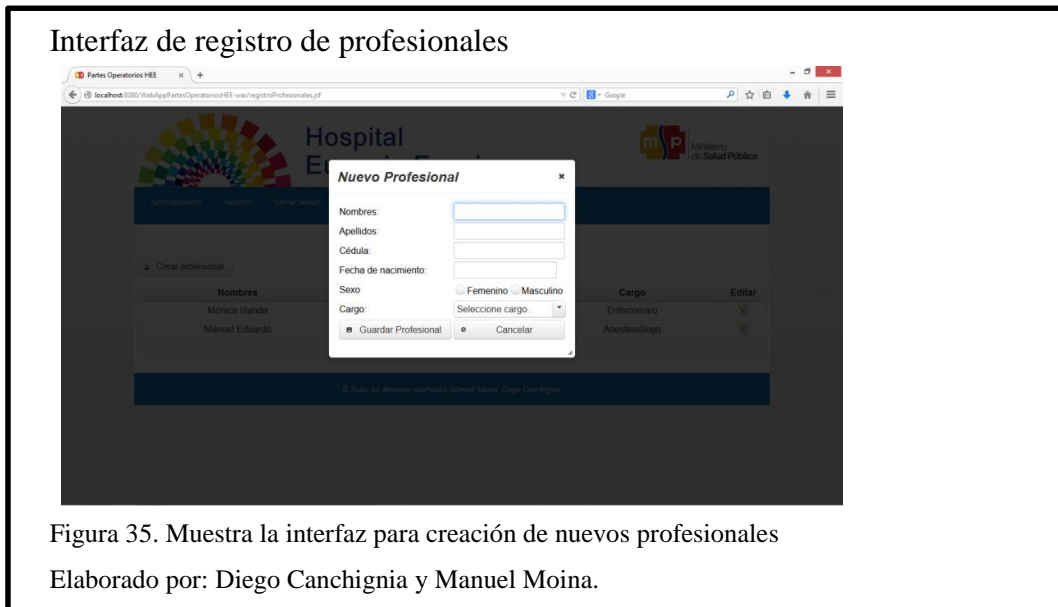
3.3.3.1. Página principal de registro de profesionales



Figura 34. Muestra el menú de registro de profesionales

Elaborado por: Diego Canchignia y Manuel Moina.

3.3.3.2. Registro de profesionales.



```
public void btnGuardarProfesional(){ boolean flagPersona=
personaFacade.agregarPersona(nuevoProfesional.getIdpersona());
    if(flagPersona){
        boolean flag = profesionalFacade.agregarProfesional(nuevoProfesional);
        if(flag){ FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_INFO, "Profesional", "
guardado correctamente."));
            nuevoProfesional = new Profesional();
            nuevoProfesional.setIdcargo(new Cargo());
            nuevoProfesional.setIdpersona(new Persona());
            listaProfesionales = profesionalFacade.getProfesionales();
            RequestContext.getCurrentInstance().execute("dlgCrearProfesional.hide()");
        } else{ FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Profesional", "no
se guardo."));
        } }else{ FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Profesional", "no
se guardo.")); } }
```

3.3.4. Pedido de cirugía

3.3.4.1. Página principal de pedido de cirugía

Interfaz principal de pedido de cirugía



Paciente	H.C.	Sexo	Edad	Servicio	Piso	Área	Cama
MOINA CAMPOS MANUEL EDUARDO	827654	Femenino	27	UCI	1	area 1	12
MASABANDA CHICAZA GERMANIA NARCISA	845462	Masculino	38	Cirugia General	1	area 1	12
VERA BRAVO PATRICIA MELANIA	827656	Masculino	23	Cirugia General	1	area 1	12
ARIAS GALLO MARGIT VERONICA	826545	Masculino	40	Neurologia Clinica	1	area 112	12
ENDARA ALPUSIG MARIA OBIDULIA	812211	Masculino	73	Cirugia General	1	Area de prueba	2

Figura 36. Muestra el menú de pedido de cirugía

Elaborado por: Diego Canchignia y Manuel Moina.

3.3.4.2. Registro del pedido de cirugía

Interfaz de registro de pedido de cirugía

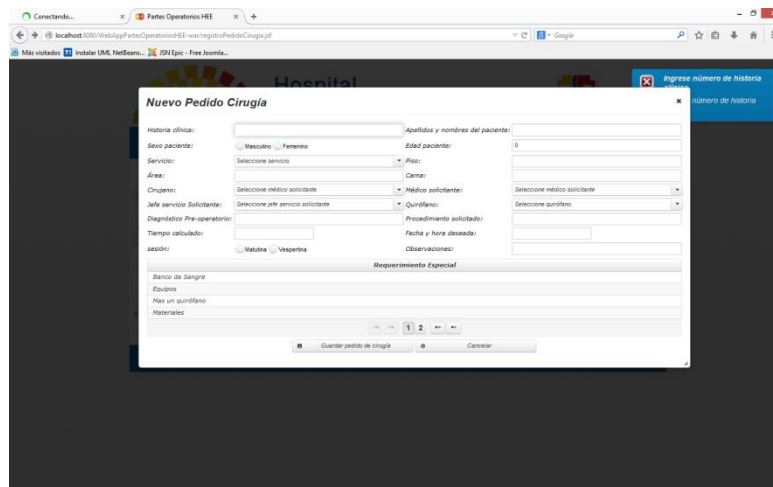


Figura 37. Muestra la interfaz para creación de un nuevo pedido de cirugía

Elaborado por: Diego Canchignia y Manuel Moina.

```
public void btnGuardarPedidoCirugia(Usuario usuario){ Estado estado = new Estado();
```



```

estado = estadoFacade.find(1); nuevoPedidoCirugia.setIdusuario(usuario);
nuevoPedidoCirugia.setIdestado(estado);
boolean flag = pedidocirugiaFacade.guardarPedidoCirugia(nuevoPedidoCirugia);
if(flag){
    guardarRequerimientosxParteOperatorio(nuevoPedidoCirugia,
    listaRequerimientosEspecialesSeleccionados);
    nuevoPedidoCirugia = new Pedidocirugia();
    nuevoPedidoCirugia.setIdquirofano(new Quirofano());
    nuevoPedidoCirugia.setIdestado(new Estado());
    nuevoPedidoCirugia.setIdcirujano(new Profesional());
    listaPedidoCirugia = pedidocirugiaFacade.findAll();
FacesContext.getCurrentInstance().addMessage(null,newFacesMessage(FacesMessage.
SEVERITY_INFO, "Pedido de cirugía ", "guardado
correctamente.));RequestContext.getCurrentInstance()
.execute("dlgCrearPedidoCirugia.hide()");}else{ FacesContext.getCurrentInstance().add
Message(null,new FacesMessage(FacesMessage.SEVERITY_ERROR, "Pedido de
cirugía ", "no se guardo.)); } }

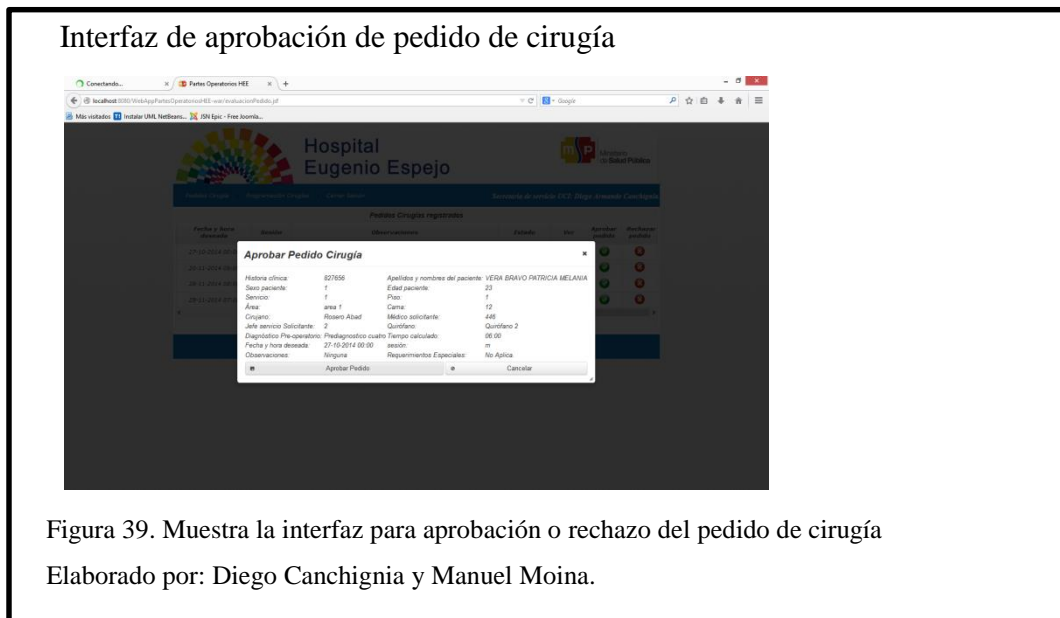
```

3.3.5. Aprobación de Solicitud de Cirugía

3.3.5.1. Pantalla principal de solicitud de cirugía



3.3.5.2. Aprobación de solicitud de cirugía



La aprobación de la solicitud dependerá de la disponibilidad de quirófanos, profesionales de la salud, implementos e insumos médicos, bajo el criterio de los responsables.

```
public void btnAprobarPedidoCirugia(Usuario usuario){  
    Estado estado = new Estado(2);  
    pedidoCirugiaSeleccionado.setIdestado(estado);  
    boolean flag =  
    pedidocirugiaFacade.modificarPedidoCirugia(pedidoCirugiaSeleccionado);  
    if(flag){  
        FacesContext.getCurrentInstance().addMessage(null,  
            new FacesMessage(FacesMessage.SEVERITY_INFO, "Pedido de cirugía ",  
                "aprobado correctamente."));  
        RequestContext.getCurrentInstance()  
            .execute("dlgAprobarPedidoCirugia.hide()");  
        estado = estadoFacade.find(1);  
        listaPedidos = pedidocirugiaFacade.getListaPedidos(estado,  
            usuario.getIdservicio());  
    }  
}
```

```

    }else{
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Pedido de cirugía
", "no se aprobo."));    }    }

```

Caso contrario se rechaza el pedido de la cirugía, y se ve reflejado en el perfil de secretaría de especialidad.

```

public void btnAprobarPedidoCirugia(Usuario usuario){
    Estado estado = new Estado(2);
    pedidoCirugiaSeleccionado.setIdestado(estado);
    boolean                                flag                                =
    pedidocirugiaFacade.modificarPedidoCirugia(pedidoCirugiaSeleccionado);
    if(flag){
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_INFO, "Pedido de cirugía ",
"aprobado correctamente."));
        RequestContext.getCurrentInstance()
            .execute("dlgAprobarPedidoCirugia.hide()");
        estado = estadoFacade.find(1);
        listaPedidos                                =                                pedidocirugiaFacade.getLitaPedidos(estado,
usuario.getIdservicio());
    }else{
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage(FacesMessage.SEVERITY_ERROR, "Pedido de cirugía
", "no se aprobo."));    }    }

```

CAPÍTULO 4

IMPLANTACIÓN

4.1. Pruebas de seguridad de acceso

4.1.1. Inyección SQL

Se verificó que no existe ninguna coincidencia al realizar la consulta `SELECT * FROM usuarios WHERE usuario = 'mmoina' AND clave = 'mmoina'`, en el gestor de base de datos ya que la clave está encriptado bajo MD5 y SHA1.

Consulta en la base de datos con usuario

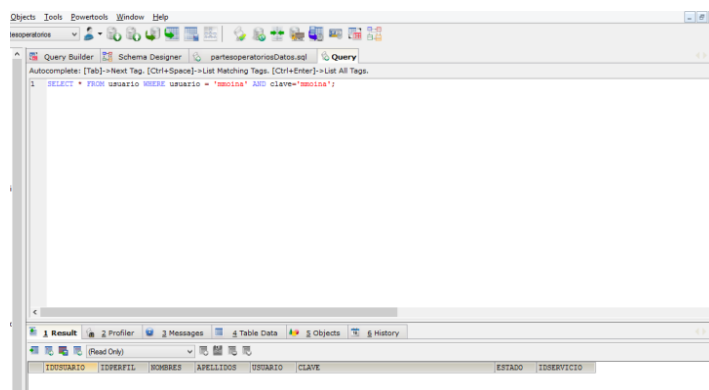


Figura 40. Muestra la consulta en la tabla usuarios sin encontrar coincidencia válida

Elaborado por: Diego Canchignia y Manuel Moina.

Se verifica en el gestor de base de datos que mediante una consulta que agregue un operador OR verdadera, dentro de la cláusula WHERE obliga a devolver un resultado sin conocer los datos del usuario y clave.

Inyección SQL en base de datos con usuario

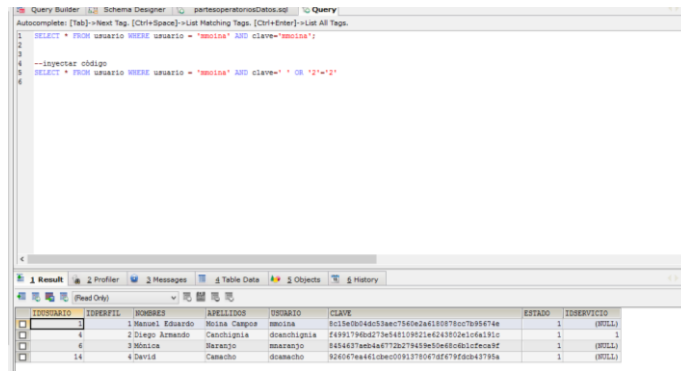


Figura 41. Muestra la consulta en la tabla usuarios, el operador OR válido fuerza a mostrar información pero la clave cumple con la encriptación

Elaborado por: Diego Canchignia y Manuel Moina.

Una vez realizada las pruebas en el gestor de la base de datos, se realiza inyección SQL en la aplicación web, sin tener un acceso.

Inyección SQL en la aplicación web

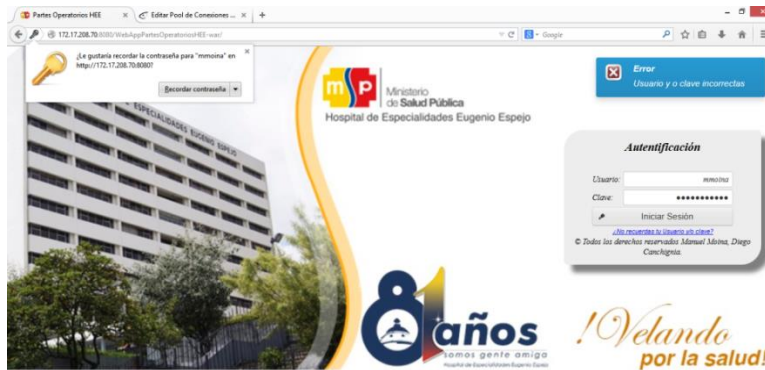


Figura 42. Muestra la inyección SQL para intentar burlar la seguridad de la aplicación

Elaborado por: Diego Canchignia y Manuel Moina.

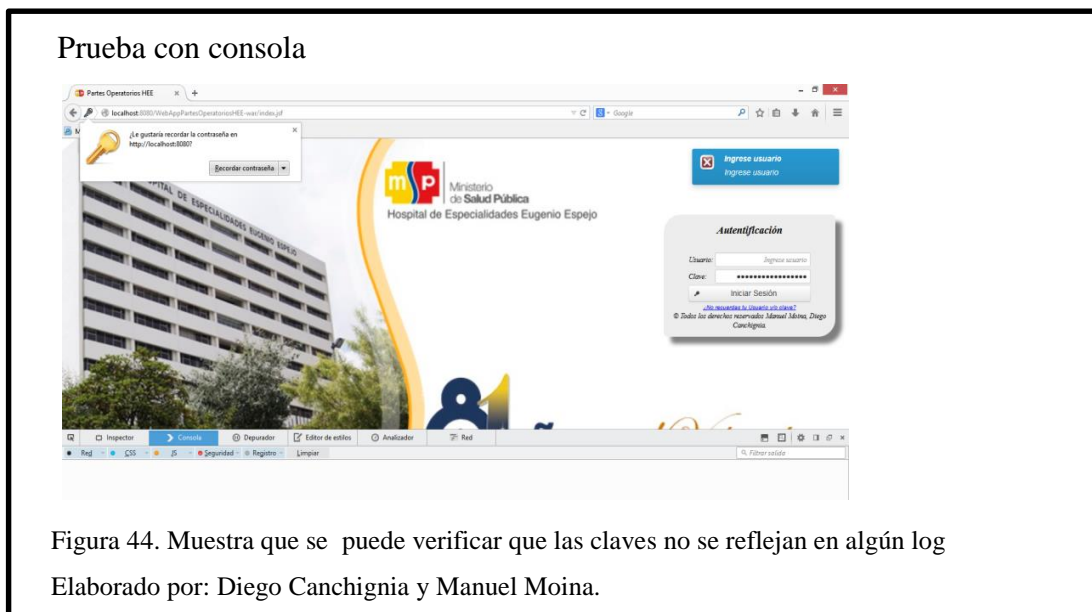
4.1.2. Consola web

Hay que tener en cuenta que los navegadores web cuentan con herramientas para desarrollo web, los cuales sirven para ver el código HTML, para intentar captar algún parámetro o valor que contenga alguna petición al servidor de aplicaciones.

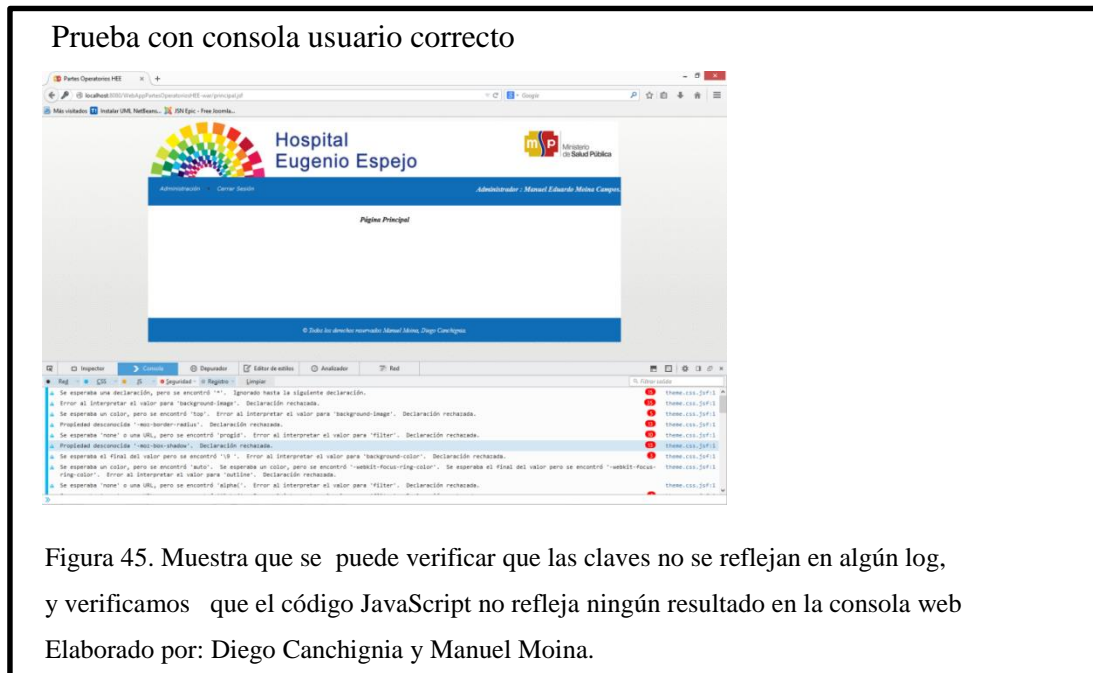
Se realiza pruebas a la aplicación web, para verificar que no haya alguna brecha de seguridad. La prueba se realiza con la consola web de Firefox con la herramienta web inspector.



La verificación de seguridad al ejecutar código JavaScript se la realiza mediante la consola dentro de las herramientas de desarrollo web de Firefox.



Al ingresar a la página principal con un usuario correcto se despliega notificaciones en la consola que se refieren a las hojas de estilo del tema de primefaces en este caso Bootstrap



4.2. Pruebas de Funcionalidad y Rendimiento

4.2.1. Funcionalidad

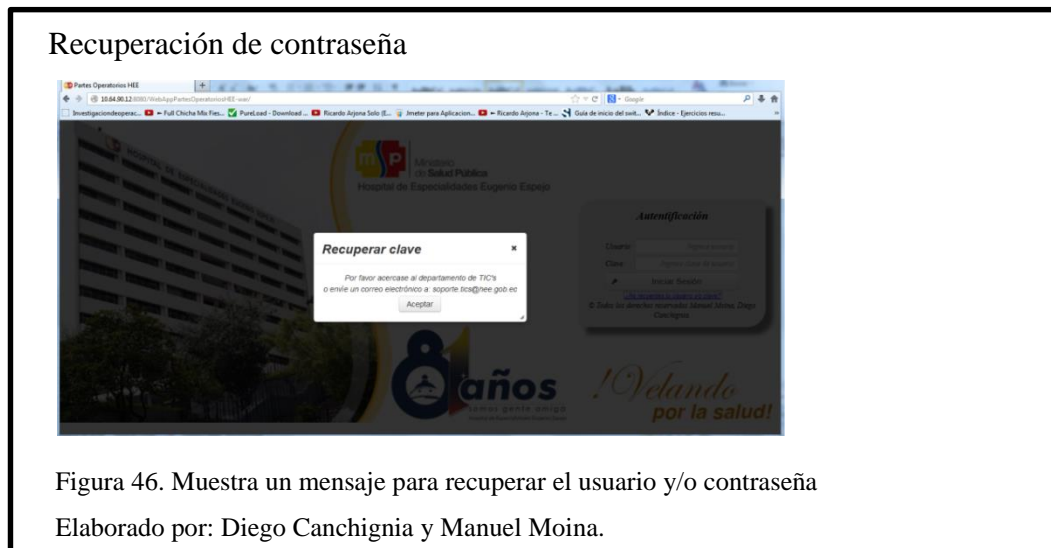
4.2.1.1. Plan de Pruebas 1

Pruebas para la autenticación de usuarios:

- Ingresar usuario y contraseña correcta
- Ingresar usuario correcto y contraseña incorrecta
- Ingresar usuario incorrecto y contraseña correcta
- Validar usuarios logeado
- Recuperar contraseña

Verificación de pruebas:

- Revisar figura 47 (Prueba con consola usuario correcto). Prueba exitosa, el usuario ingresa a la página principal
- Revisar figura 42 (Inyección SQL en la aplicación web). Prueba exitosa, el usuario no ingresa a la página principal, muestra una advertencia que el usuario y/o contraseña son incorrectas
- Revisar figura 30 (Interfaz de control de acceso). Prueba exitosa, el usuario no ingresa a la página principal, el filtro valida que el usuario está logeado, caso contrario le regresa a la pantalla de autenticación
- Revisar figura 46 (Recuperación de contraseña). Prueba exitosa, el usuario olvido su contraseña, y se envía un link al correo registrado en el registro de usuario



A continuación se muestra una tabla con el plan de pruebas 1 que mejora la visualización de las pruebas ejecutadas.

Tabla 19.

Plan de pruebas 1

Plan de Pruebas 1			
Pruebas para la autenticación de usuarios	Funciona		Validación
	Sí	No	
Ingresar usuario y contraseña correcta.	X		El usuario ingresa a la página principal
Ingresar usuario correcto y contraseña incorrecta.	X		El usuario no ingresa a la página principal, muestra una advertencia que el usuario y/o contraseña son incorrectas
Ingresar usuario incorrecto y contraseña correcta.	X		El usuario no ingresa a la página principal, muestra una advertencia que el usuario y/o contraseña son incorrectas
Validar usuarios logeado.	X		El usuario no ingresa a la página principal, el filtro valida que el usuario está logeado, caso contrario le regresa a la pantalla de autenticación.
Recuperar contraseña.	X		El usuario olvido su contraseña, y se envía un link al correo registrado en el registro de usuario

Nota: muestra las pruebas ejecutadas para la autenticación de usuarios

Elaborado por: Diego Canchignia y Manuel Moina.

4.2.1.2. Plan de Pruebas 2

Pruebas para registro de usuarios:

- Ingresar como administrador de la aplicación
- Registrar nuevo usuario
- Ingresar con usuario registrado
- Verificar página principal del usuario

Verificación de pruebas:

- Revisar figura 47 (Administrador de la aplicación). Prueba correcta, se ingresa con cuenta de administrador para crear un nuevo usuario.

Administrador de la aplicación



Figura 47. Muestra la pantalla principal del administrador de la aplicación
Elaborado por: Diego Canchignia y Manuel Moina.

- Revisar figura 48 (Nuevo usuario). Prueba exitosa, se registra el nuevo usuario.

Nuevo usuario

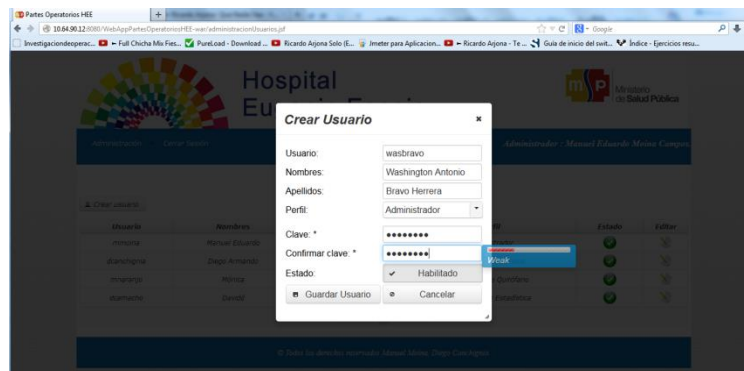


Figura 48. Muestra el registro de un nuevo usuario, para la aplicación web y se le asigna un perfil
Elaborado por: Diego Canchignia y Manuel Moina.

- Revisar figura 49 (Ingreso con usuario nuevo). Prueba exitosa, accede el nuevo usuario.

Ingreso con usuario nuevo



Figura 49. Muestra la página principal a la que puede acceder el nuevo usuario registrado

Elaborado por: Diego Canchignia y Manuel Moina.

A continuación se muestra una tabla con el plan de pruebas 2 que mejora la visualización de las pruebas ejecutadas.

Tabla 20.

Plan de pruebas 2

Plan de Pruebas 2			
	Funciona		
Pruebas para registro de usuarios	Sí	No	Validación
Ingresar como administrador de la aplicación.	X		Se ingresa con cuenta de administrador para crear un nuevo usuario
Registrar nuevo usuario	X		Se ingresa con cuenta de administrador para crear un nuevo usuario
Registrar nuevo usuario	X		Se registra el nuevo usuario
Ingresar con usuario registrado	X		Accede el nuevo usuario
Verificar página principal del usuario	X		Accede el nuevo usuario y revisa la página principal

Nota: muestra las pruebas ejecutadas para el registro de usuarios

Elaborado por: Diego Canchignia y Manuel Moina.

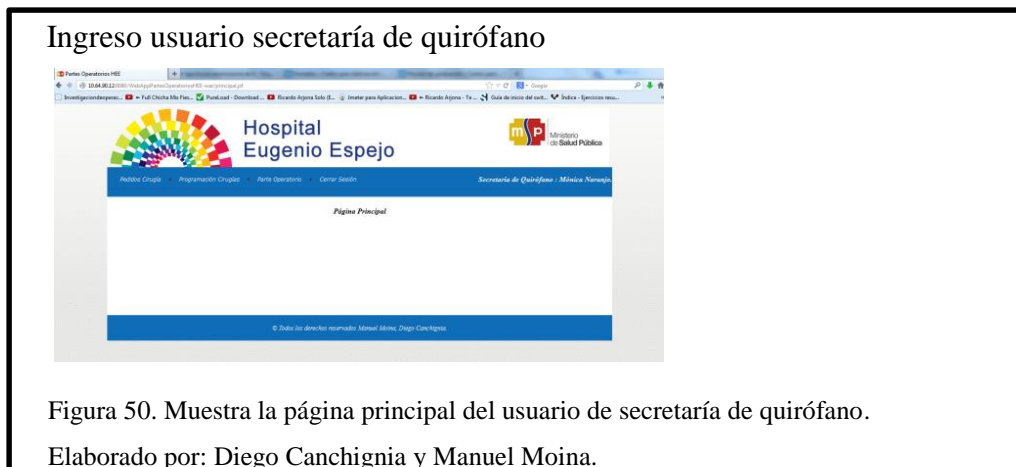
4.2.1.3. Plan de Pruebas 3

Pruebas para registro de Partes Operatorios:

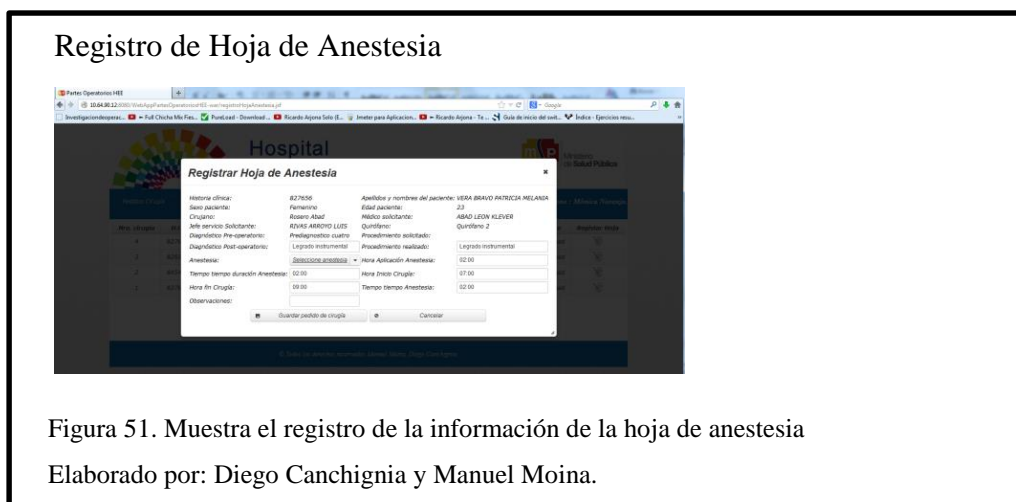
- Ingresar con el usuario con perfil de secretaría de quirófano
- Registrar Hoja de Anestesia
- Verificar registro

Verificación de Pruebas:

- Revisar figura 50 (Ingreso usuario secretaría de quirófano), Prueba exitosa, el usuario ingresa con el perfil para registrar el parte operatorio.



- Revisar figura 51 (Registro de Hoja de Anestesia). Prueba exitosa, el usuario registra la información.



- Revisar figura 52 (Verificación del registro del parte operatorio). Prueba exitosa, muestra el registro del parte operatorio.

Verificación del registro del parte operatorio

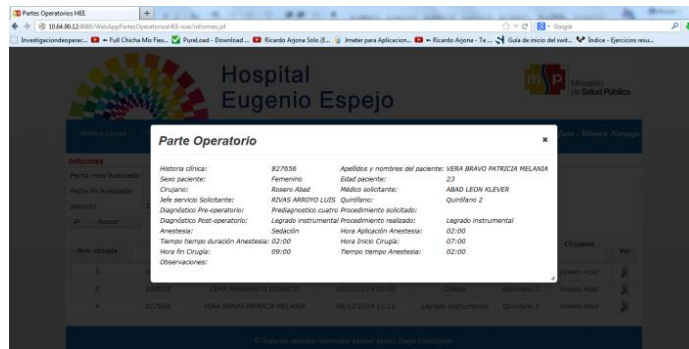


Figura 52. Muestra el registro de la información del parte operatorio
Elaborado por: Diego Canchignia y Manuel Moina.

A continuación se muestra una tabla con el plan de pruebas 3 que mejora la visualización de las pruebas ejecutadas.

Tabla 21.

Plan de pruebas 3

Plan de Pruebas 3			
Pruebas para registro de Partes Operatorios	Funciona		Validación
	Sí	No	
Ingresar con el usuario con perfil de secretaría de quirófano	X		El usuario ingresa con el perfil para registrar el parte operatorio
Registrar Hoja de Anestesia	X		El usuario registra la información
Verificar registro	X		Se registra el parte operatorio

Nota: Muestra las pruebas ejecutadas para el registro de Partes Operatorios

Elaborado por: Diego Canchignia y Manuel Moina.

4.2.2. Rendimiento

Las pruebas de rendimiento se la realizo con Jmeter, software que es propiedad de Apache, el cual sirve para analizar el rendimiento de una aplicación web.

Los parámetros de pruebas de rendimiento se los configura de la siguiente manera:

- Escoger Plan de Pruebas
- Añadir el Grupo de Hilos
- Añadir Controlador Simple
- Añadir Petición HTTP, para configurar los parámetros del servidor y puerto

➤ Añadir Ver Visor de Respuesta

Jmeter petición HTTP

Figura 53. Muestra la configuración de servidor al que se apunta, y la ruta de la aplicación web

Elaborado por: Diego Canchignia y Manuel Moina.

Una de las pruebas que se puede realizar en Jmeter es observar el número de la muestra, la hora de inicio y los parámetros para evaluar el rendimiento de la aplicación web.

Jmeter Ver Resultado en Árbol

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra	Estado	Bytes	Latency
1	20 14 14.095	Grupo de Hilos 1-1	Petición HTTP	44	OK	4847	44
2	20 14 14.140	Grupo de Hilos 1-1	Petición HTTP	25	OK	4845	25
3	20 14 14.187	Grupo de Hilos 1-1	Petición HTTP	39	OK	4845	39
4	20 14 14.207	Grupo de Hilos 1-1	Petición HTTP	25	OK	4847	25
5	20 14 14.233	Grupo de Hilos 1-1	Petición HTTP	24	OK	4845	24
6	20 14 14.259	Grupo de Hilos 1-1	Petición HTTP	25	OK	4845	25
7	20 14 14.285	Grupo de Hilos 1-1	Petición HTTP	76	OK	4845	76
8	20 14 14.365	Grupo de Hilos 1-1	Petición HTTP	57	OK	4845	57
9	20 14 14.397	Grupo de Hilos 1-2	Petición HTTP	48	OK	4847	48
10	20 14 14.425	Grupo de Hilos 1-1	Petición HTTP	35	OK	4845	35
11	20 14 14.447	Grupo de Hilos 1-2	Petición HTTP	49	OK	4845	49
12	20 14 14.452	Grupo de Hilos 1-1	Petición HTTP	48	OK	4847	47
13	20 14 14.497	Grupo de Hilos 1-2	Petición HTTP	25	OK	4845	25
14	20 14 14.511	Grupo de Hilos 1-1	Petición HTTP	24	OK	4845	24
15	20 14 14.523	Grupo de Hilos 1-2	Petición HTTP	24	OK	4845	24
16	20 14 14.537	Grupo de Hilos 1-1	Petición HTTP	21	OK	4845	21
17	20 14 14.549	Grupo de Hilos 1-2	Petición HTTP	20	OK	4845	20
18	20 14 14.560	Grupo de Hilos 1-1	Petición HTTP	21	OK	4847	21
19	20 14 14.571	Grupo de Hilos 1-2	Petición HTTP	20	OK	4847	20
20	20 14 14.582	Grupo de Hilos 1-1	Petición HTTP	21	OK	4845	21
21	20 14 14.593	Grupo de Hilos 1-2	Petición HTTP	24	OK	4845	24
22	20 14 14.595	Grupo de Hilos 1-1	Petición HTTP	24	OK	4847	24
23	20 14 14.619	Grupo de Hilos 1-2	Petición HTTP	25	OK	4847	25
24	20 14 14.630	Grupo de Hilos 1-1	Petición HTTP	25	OK	4847	25
25	20 14 14.645	Grupo de Hilos 1-2	Petición HTTP	23	OK	4847	23
26	20 14 14.656	Grupo de Hilos 1-1	Petición HTTP	19	OK	4845	19
27	20 14 14.668	Grupo de Hilos 1-2	Petición HTTP	20	OK	4845	20

Figura 54. Muestra el resultado de la prueba de rendimiento, se puede observar el número de muestra, la hora de inicio, la etiqueta, el estado, los Bytes y la latencia, estos datos nos indican que el rendimiento cuenta con parámetros dentro del rendimiento normal de la aplicación

Elaborado por: Diego Canchignia y Manuel Moina.

Este resultado se puede observar en la figura, que se encuentra a continuación.

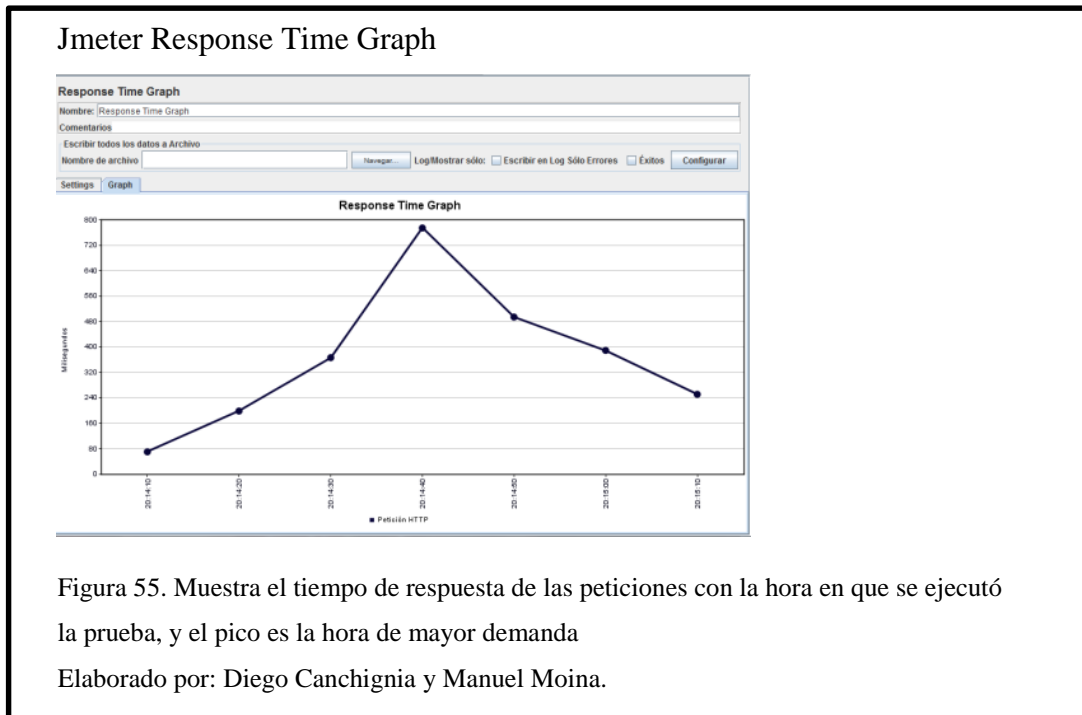


Figura 55. Muestra el tiempo de respuesta de las peticiones con la hora en que se ejecutó la prueba, y el pico es la hora de mayor demanda

Elaborado por: Diego Canchignia y Manuel Moina.

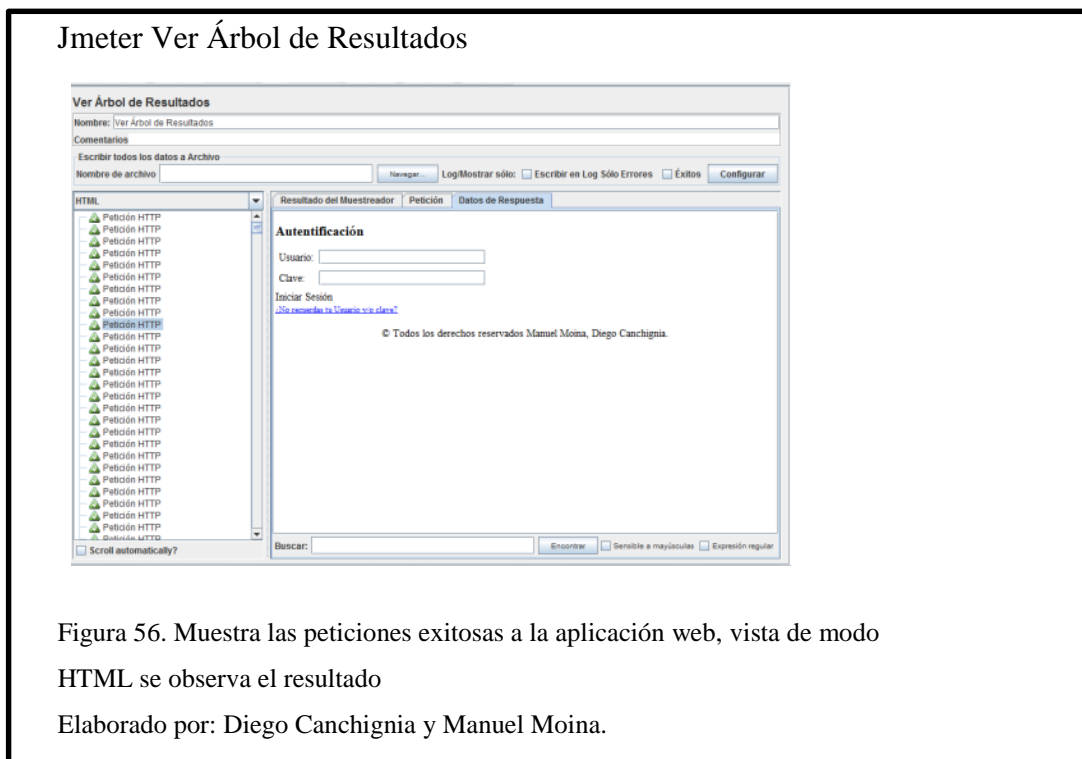


Figura 56. Muestra las peticiones exitosas a la aplicación web, vista de modo HTML se observa el resultado

Elaborado por: Diego Canchignia y Manuel Moina.

Jmeter Figuras de Resultados



Figura 57. Muestra el gráfico del rendimiento con los resultados de los datos, media, mediana, desviación y rendimiento, observamos un resultado dentro de los parámetros normales en referencia del número de muestras
Elaborado por: Diego Canchignia y Manuel Moina.

Este resultado se puede observar en la figura, que se encuentra a continuación

Jmeter Visualizador Spline

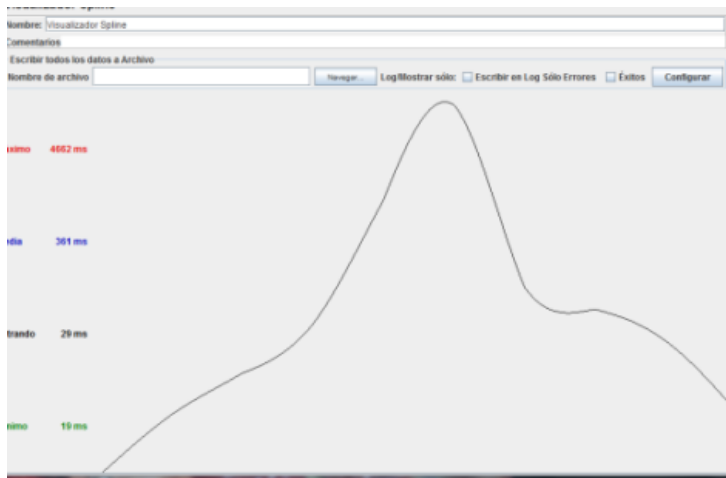


Figura 58. Muestra los datos obtenidos en los gráficos de resultados, pero en un lapso de tiempo en milisegundos, se observa que los valores de los datos en un máximo y un mínimo, se puede verificar una gráfica similar a la del Response Time Graph.
Elaborado por: Diego Canchignia y Manuel Moina.

Jmeter Gráfico

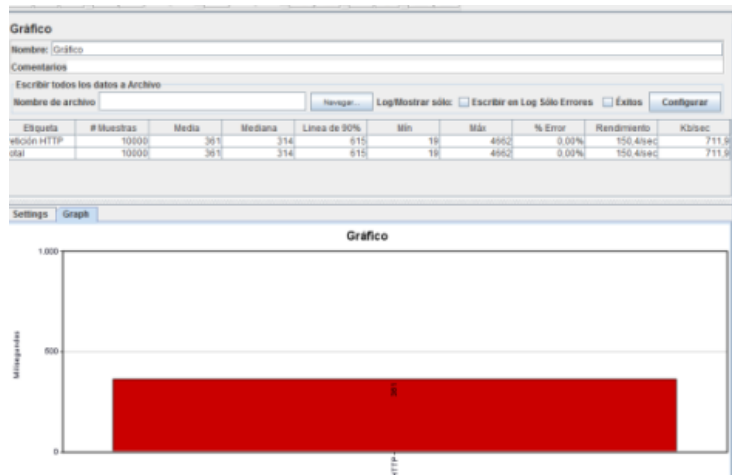


Figura 59. Muestra los valores totales del gráfico de resultado

Elaborado por: Diego Canchignia y Manuel Moina.

4.3. Implantación

Para garantizar que la implantación de la aplicación web sea exitosa y procurando un impacto bajo, teniendo en cuenta que al iniciar con una aplicación se poder complicar con los actores directos e indirectos de dicha aplicación, se recomienda con los siguientes puntos:

- Coordinar con la Subdirección de Docencia e Investigación para que realice los trámites pertinentes con el Departamento de TIC'S para la planificación de la implantación
- Realizar un plan de contingencia conjuntamente con el Departamento de TIC'S, contando previamente con la planificación de la implantación, en caso de una falla eventual del sistema, o hasta estabilizar su funcionamiento
- Comenzar con la utilización del sistema piloto con una especialidad que tengo un margen de cirugías modesta o baja, y paulatinamente acoplar las demás especialidades
- Monitorear en funcionamiento y rendimiento de la aplicación web, verificar los tiempos de peticiones estén dentro del margen aceptable y comprobado en las pruebas ejecutadas

- Prever en días no laborales, o de menor concurrencia de cirugías la modificación o ajustes en la aplicación web
- Entregar dos copias del código, una para modificaciones y la otra como respaldo en caso de falla en los ajuste

CONCLUSIONES

- La aplicación web lleva un registro de información ordenada y detallada de los partes operatorios, esto reduce el tiempo en las búsquedas en el área de quirófanos y las diferentes secretarías de especialidades del Hospital de Especialidades Eugenio Espejo
- Con el uso del sistema se agiliza la visualización de las fechas y horas de las cirugías quirúrgicas, tanto para los pacientes como a los profesionales de la salud
- Con la implementación del agendamiento de cirugías se evita la pérdida de documentos y duplicación de información que puede surgir para las intervenciones quirúrgicas, lo que proporciona un mejor servicio en las programaciones para los quirófanos de esta casa de salud
- Al disponer de bitácoras, el hospital dispone de todos los registros de las intervenciones quirúrgicas, con el fin de auditar y validar todos los procedimientos efectuados
- Los Figuras estadísticos permiten a otras áreas de la institución generar informes e indicadores de desempeño
- La aplicación web está diseñada para acoplarse a otros módulos existentes en la casa de salud
- Al ser una aplicación web, no afecta la plataforma de sistema operativo que ocupe el usuario, solo se necesita de un navegador web
- Con la utilización de perfiles y control de acceso, se salvaguarda la información registrada en el parte operatorio

RECOMENDACIONES

- Al contar con una Base de Datos, se aconseja contar con políticas para realizar backup en un lapso de tiempo
- La institución al contar con un Firewall y un Proxy para seguridades, deben habilitar los puertos 3306 para la Base de Datos y 8080 para el servidor de aplicaciones web
- Los usuarios del sistema deben contar con un navegador web en la versión 17 o superior de Mozilla Firefox, ya que la aplicación desarrollada está optimizada para este navegador
- Sí a futuro se requiere de nuevos módulos para la aplicación web se recomienda planificar con los distintos servicios de la institución las necesidades y prioridades a desarrollarse
- Al realizar este proyecto se observa que es de vital importancia el levantamiento de los requerimientos, lo que implica tomarse un tiempo prudente para ello, esto ayuda a tener claro lo que el cliente requiere, y se facilita el desarrollo de la aplicación

LISTA DE REFERENCIAS

- Barriga, Diego Paúl Tamayo. (2014). *Respositorios de Tesis*. Recuperado el Septiembre de 2014, de dspace.espech.edu.ec:
<http://repositorio.espe.edu.ec/bitstream/21000/4431/1/M-ESPEL-0017.pdf>
- Basta, S. G. (2012). *Red Tercer Milenio*. Recuperado el Abril de 2014, de Libro digital - Metodologia de la investigacion:
http://www.aliatuniversidades.com.mx/bibliotecasdigitales/pdf/axiologicas/Metodologia_de_la_investigacion.pdf
- Bernd Bruegge, A. H. (2002). *Ingenieria de Software Orientada a Objetos* (Primera Edición ed.). México: Prentice-Hall.
- bestoutcome. (6 de Diciembre de 2005). *Agile and scrum*. Recuperado el 26 de Agostos de 2014, de <http://www.bestoutcome.com/overview-of-scrum.html>
- Bestoutcome. (6 de Diciembre de 2005). *Agile and scrum*. Recuperado el 26 de Agostos de 2014, de <http://www.bestoutcome.com/overview-of-scrum.html>
- Blogspot. (Abril de 2010). *Blogspot*. Recuperado el Septiembre de 2014, de <http://demetole.blogspot.com/2010/04/glassfish-o-tomcat-cual-le-conviene.html>
- Blogspot. (Abril de 2010). *Web Blogspot*. Recuperado el Septiembre de 2014, de <http://demetole.blogspot.com/2010/04/glassfish-o-tomcat-cual-le-conviene.html>
- Byron Alejandro Andrango Correa, D. F. (Junio de 2013). *Repositorio de Tesis Salesiana*. Recuperado el Junio de 2014, de dspace.ups.edu.ec/bitstream/123456789/5184/1/UPS-ST001012.pdf
- Christian Jonathan Bermeo Astudillo. (Junio de 2012). *Repositorio Tesis Salesiana*. Recuperado el Septiembre de 2014, de <http://dspace.ups.edu.ec/bitstream/123456789/3077/1/UPS-CT002500.pdf>

Davenport, T. (2000). *The knowledge management toolkit: practical techniques for building a knowledge management system*. NJ, USA: Prentice Hall PTR Upper Saddle River.

Departamento de Estadística Hospital de Especialidades Eugenio Espejo. (2014). *Estadística de pacientes mensuales 2014*. Quito.

Departamento de Estadística Hospital de Especialidades Eugenio Espejo. (2014). *Estadística de pacientes mensuales HEE 2014*. Quito.

Departamento de Talento Humano Hospital de Especialidades Eugenio Espejo. (2014). *Nomina Abril de empleados HEE 2014*. Quito.

Departamento de Talento Humano Hospital de Especialidades Eugenio Espejo. (2014). *Nomina Abril 2014*. Quito.

Enrique Viñé Lerma. (Junio de 2010). *Adictos al trabajo*. Recuperado el Septiembre de 2014, de <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=introduccionPrimefaces>

Hitt, B. y. (2006). *Administración*. México: Pearson.

Hospital de los Andes. (Abril de 2012). *REGISTRO DE EVALUACIÓN PREANESTÉSICA*. Recuperado el Abril de 2014, de biblioteca virtual: http://hospitaldelosandes.cl/biblioteca_virtual/wp-content/uploads/2013/05/EVAL.PREANESTESICA-PROT.ANESTESIA-Y-PROT.OPERATORIO-Final.pdf

Hospital Eugenio Espejo. (2013). *Página del Hospital Eugenio Espejo*. Obtenido de <http://www.hee.gob.ec/webhee2013/index.php/el-hospital/historia>

Hospital Eugenio Espejo. (2013). *Página web oficial del Hospital Eugenio Espejo*. Obtenido de <http://www.hee.gob.ec/webhee2013/index.php/el-hospital/historia>

Ideasoft . (2007). *Ideasoft*. Recuperado el Abril de 2014, de Introducción a JMS:
<http://www.fing.edu.uy/inco/cursos/tsi/TSI3/TSI3-2007-JMS.pdf>

JBOSS. (Marzo de 2014). *Docs Jboss*. Recuperado el Septiembre de 2014, de
<https://docs.jboss.org/author/download/attachments/66322959/JBossAS7-JavaEE.png?version=1&modificationDate=1310399117000>

LINS. (11 de Septiembre de 2012). <http://www.fing.edu.uy>. Recuperado el 11 de Abril de 2014, de Documento digital - Taller de sistema de Información 2:
<http://www.fing.edu.uy/inco/cursos/tsi/TSI2/2012/teorico/tsi2-07-web-services.pdf>

Luis Joyanes Aguilar, I. Z. (2010). *C, C++, JAVA y UML* (Primera ed.). México: McGraw-Hill.

Microsoft Corporation. (Diciembre de 2006). www.microsoft.com/soa. Recuperado el Abril de 2014, de www.microsoft.com/soa:
http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&uact=8&ved=0CEEQFjAF&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2Fc%2F2%2Fc%2Fc2ce8a3a-b4df-4a12-ba18-7e050aef3364%2F070717-Real_World_SOA.pdf&ei=6o5MU_TIH8670AHJ-oHQDQ&usg=

Molina Segovia, J. E. (Septiembre de 2008). *Escuela Politécnica Nacional*. Recuperado el Abril de 2014, de Repositorio Digital Politécnico-Tesis Shaman:
<http://bibdigital.epn.edu.ec/handle/15000/924>

Oracle. (2013). *docs.oracle.com*. Recuperado el Abril de 2014, de Documentos digitales - Referencias técnicas: <http://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html>

Oracle. (2013). <http://docs.oracle.com/>. Recuperado el Abril de 2014, de Documentos digitales - Referencias técnicas:
<http://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html>

- Oracle. (Enero de 2014). *http://www.java.com/*. Recuperado el Abril de 2014, de Especificaciones: *http://www.java.com/es/download/faq/techinfo.xml*
- Oracle. (Enero de 2014). *web java.com*. Recuperado el Abril de 2014, de Especificaciones: *http://www.java.com/es/download/faq/techinfo.xml*
- Oscar Steve Castro Arteaga. (2013). *Repositorio Tesis UPS*. Recuperado el Septiembre de 2014, de Repositorio Sistemas Cuenca: *http://dspace.ups.edu.ec/handle/123456789/3584*
- Oviedo Guallichico Danilo Roman, Granda León José Antonio. (Febrero de 2014). *Repositorio de Tesis Salesiana*. Recuperado el Setiembre de 2014, de *http://dspace.ups.edu.ec/bitstream/123456789/6362/1/UPS-ST001090.pdf*
- Pereira, U. T. (2007). *http://repositorio.utp.edu.co*. Recuperado el 2014, de *http://repositorio.utp.edu.co/dspace/bitstream/11059/791/1/0053L155mc.pdf*
- Softeng Software Engineers. (Enero de 2013). *http://www.softeng.es*. Recuperado el Abril de 2014, de Metodologia scrum para el desarrollo de software - Aplicaciones complejas: *http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html*
- Softeng Software Engineers. (Enero de 2013). *web softeng.es*. Recuperado el Abril de 2014, de Metodologia scrum para el desarrollo de software - Aplicaciones complejas: *http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html*
- University of Alicante. (Febreo de 2004). *Web jtech.ua.es*. Recuperado el 24 de Septiembre de 2014, de *http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/ejb/sesion01-apuntes.htm*
- Victor Hernán Quimbiamba Lanchimba. (Septiembre de 2013). *Repositorio Tesis Salesiana*. Recuperado el Septiembre de 2014, de *dspace.ups.edu.ec/bitstream/123456789/5345/1/UPS-ST001028.pdf*

Wehrich, H. K. (1999). *Administración: Una perspectiva global*. México: McGraw-Hill.